Superlinear Deterministic Top-Down Tree Transducers

Ph.D. Thesis

by

Gábor Dányi

József Attila University
Departments of Informatics
Szeged, Hungary

1997

Contents

List of Figures

Introduction

In theoretical computer science tree transducers have been studied since the early seventies. They are finite devices processing terms over ranked alphabets. Such terms are called trees in this area. A tree transducer induces a binary relation over tree sets, called a tree transformation.

Several types of tree transducers have been defined. Namely, the concept of the top-down tree transducer was introduced in [Rou] and [Tha1]. Then the notion of the bottom-up tree transducer was defined in [Tha2]. Later on, in order to increase the transformational capacity, more powerful devices were introduced, such as top-down tree transducers with regular look-ahead (see [Eng2]), macro tree transducers (see [EngVog1]), attributed tree transducers (see [Fül1]), high level tree transducers (see [EngVog2]), modular tree transducers (see [EngVog3]), and high level modular tree transducers (see [Vog]).

In this thesis we shall consider only deterministic top-down tree transducers and tree transformations induced by such tree transducers.

The motivation of studying top-down tree transducers is that they serve as formal models of syntax-directed compilers, thus tree transformations induced by top-down tree transducers are abstract models of translations realized by syntax-directed compilers, see [Eng4].

Several restricted subtypes of deterministic top-down tree transducers have been defined and investigated. In this thesis we work with, among others, total, linear, nondeleting, and homomorphism deterministic top-down tree transducers.

In our sense, a tree transformation class is generally a class of tree transformations induced by tree transducers of a certain type. Thus we can distinguish the class of deterministic top-down tree transformations, denoted by DT, and its subclasses of total, linear, nondeleting, and homomorphism deterministic top-down tree transformations, denoted by $t\text{-}DT$, $l\text{-}DT$, $nd\text{-}DT$, and HOM, respectively. Moreover, type properties can be combined resulting more special devices. For instance, we can speak about linear and nondeleting deterministic top-down tree transducers, of which the induced tree transformation class is denoted by $l\text{-}nd\text{-}DT$.

Investigating a certain type of top-down tree transducers, the questions naturally arise, what sets of trees can be processed by top-down tree transducers of that type, and what sets of trees can occur as results of such processings. For a top-down tree transducer, the sets of possible input and output trees are called the domain and the range of the induced tree transformation, respectively.

Tree sets are also called tree languages. Similarly to string languages, for tree languages there also exist finite state recognizers. Using these devices, we can define the classes of recognizable and deterministic recognizable tree languages, see [GécSte4]. It turned out that the domains of deterministic top-down tree transformations are exactly the deterministic recognizable tree languages. Moreover, the class of ranges of linear deterministic top-down tree transformations is exactly the class of recognizable tree languages.

Since tree transformations are binary relations over tree sets, the concept of their composition, denoted by $\circ$, is clear. Moreover, the composition operation can naturally be extended to classes of tree transformations.

Compositions and decompositions of deterministic top-down tree transformation classes are of special interest, because they model consecutive applications of deterministic top-down tree transducers of certain types to tree languages in such a way that the output of a device is the input of its successor. The motivation of studying compositions comes from the fact that applying deterministic top-down tree transducers in succession can yield extra computational power in the sense that the resulting tree transformation cannot be induced in general by a single deterministic top-down tree transducer. Similarly, the investigation of decompositions is motivated by the intention that one would like to know whether a deterministic top-down tree transformation of a certain type could also be induced by the consecutive application of two or more deterministic top-down tree transducers of simpler types.

Top-down tree transducers and top-down tree transformations were studied in a large number of papers.

In pioneer works [Rou], [Tha1], [Eng1], [Eng3], [Bak1], [Bak2] and [Bak3] several restricted types (total, linear, nondeleting, etc.) were defined, the transformational power of different types were compared to each other, and some closure properties of the corresponding tree transformation classes were explored. A good survey of these results can be found in [GécSte4]. Moreover, [FülVág1] also contains important observations concerning closure properties of the class of deterministic top-down tree transformations and its subclasses.

Recognizability of domains and ranges of top-down tree transformations also have been studied very intensively, see [Rou], [Eng2], [GécSte4], [FülVág1], and [FülVág3].

The undecidability of equivalence problem of top-down tree transducers in the general case immediately follows from the result of [Gri] on the undecidability of equivalence problem of GSM's. On the other hand, it turned out that the equivalence is decidable in the deterministic case, see [Ési1] and [Zac]. Moreover, the equivalence problem were studied for some other restricted nondeterministic types in [AndBos]. The decidability of some other properties (injectivity, recognizability of the range tree set, etc.) have also been investigated, see [Ési1], [Ési2], [Fül4], and [FülGye].

Compositions and decompositions of tree transformation classes have been studied very intensively. Almost all papers regarding tree transducers contains such results and hence a huge number of decomposition and composition equations have been obtained. It was desirable to find a uniform way for researching this area. In [FülVág4] and [FülVág6] a general method was proposed for developing an algorithm, which, for an arbitrarily fixed base set of tree transformation classes, can decide the relationship concerning the inclusions between tree transformation classes obtained by composition from base classes. The method has numerous applications using different base sets of tree transformation classes, see [FülVág4], [FülVág5], [Fül2], [SluVág], and [GyeVág].

The subject of this thesis is the characterization of a new subtype of deterministic top-down tree transducers, called *superlinear deterministic top-down tree transducers*. We denote the class of superlinear deterministic top-down tree transformations by $sl\text{-}DT$. Superlinear deterministic top-down tree transducers are specialized linear deterministic top-down tree transducers and it holds that $sl\text{-}DT \subset l\text{-}DT$.

The concept of superlinear deterministic top-down tree transducers was proposed by H. Vogler during a personal communication with Z. Fülöp in 1992. It was motivated by the well known decomposition equation $DT = nd\text{-}HOM \circ l\text{-}DT$, which appeared first time in [Engl] and [Bak3]. They discussed whether $l\text{-}DT$ in the above equation can be substituted by an even more restricted subclass of DT. It was guessed that a proper subclass of $l\text{-}DT$ would be suitable, namely the class $sl\text{-}DT$ of superlinear deterministic top-down tree transducers.

In this work we investigate properties of superlinear deterministic top-down tree transducers and the corresponding tree transformation class $sl\text{-}DT$. Although the starting point of the research was the decomposition equation $DT = nd\text{-}HOM \circ sl\text{-}DT$, we have explored many other interesting results concerning superlinear deterministic top-down tree transducers and transformations. The problems we have arisen and answered were motivated by the earlier works regarding tree transducers (see, e.g., [Engl], [Eng3], [Bak3], [FülVág1], [FülVág2]). These are as follows:

- What is the relationship between $sl\text{-}DT$ and the known tree transformation classes, such as DT or $l\text{-}DT$? In other words, how does the superlinear deterministic top-down tree transducers compare with the known types regarding transformational power?
- Is the class $sl\text{-}DT$ closed under the composition?
- What kind of tree languages can be domain and range tree languages of superlinear deterministic top-down tree transformations?
- How can we characterize the compositions of $sl\text{-}DT$ with other known classes?

The outline of the thesis is the following:

Chapter 1 We introduce the notions and notations, which are necessary to understand the results. Moreover, we also recall there some earlier results used in later chapters.

Chapter 2 We investigate certain properties of superlinear deterministic top-down tree transducers.
In Section 2.1 we prove that, among others, *sl-DT* is not closed under the composition, that $DT = nd\text{-}HOM \circ sl\text{-}DT$ holds, and that there is a total linear deterministic top-down tree transformation, which cannot be induced by consecutive application of a sequence of superlinear deterministic top-down tree transducers.
In Section 2.2 we show that the domains of superlinear deterministic top-down tree transformations are exactly those tree languages, which are recognized by tree recognizers of a new restricted type, called semi-universal deterministic top-down tree recognizers. Moreover, we prove that, for a deterministic recognizable tree language recognized by a given deterministic top-down tree recognizer, it is decidable whether this tree language can be domain of a superlinear deterministic top-down tree transformation.
Finally, in Section 2.3, it turns out that the class of ranges of superlinear deterministic top-down tree transformations is exactly the class of recognizable languages.

Chapter 3 We present two hierarchy theorems concerning superlinear deterministic top-down tree transformations, which show the uniqueness of this type in the family of deterministic top-down tree transformations. Namely, it turns out that, in contrast with the earlier types (e.g. *DT*, *l-DT*, etc.), the powers of the class *sl-DT* form a proper hierarchy. Roughly speaking, this means that the more superlinear deterministic top-down tree transducers we apply in succession, the higher transformational power we can get.
In Section 3.1 we prove that the hierarchy $\{sl\text{-}DT^n \mid n \geq 1\}$ is proper. More exactly, we prove a stronger statement, namely that $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 1\}$ is a proper hierarchy, where, for a tree transformation class $\mathcal{C}$, we denote by $\mathrm{dom}(\mathcal{C})$ the class of domains of the tree transformations in $\mathcal{C}$. Note that there are very few such hierarchy theorems in the literature, see [Eng3] and [FülVág2].
In Section 3.2 we consider total superlinear deterministic top-down tree transducers. Here we show that even the hierarchy $\{t\text{-}sl\text{-}DT^n \mid n \geq 1\}$ is proper.

Chapter 4 We explore how the class *sl-DT* behaves when composing with other known tree transformation classes. These other classes are *HOM*, *l-DT*, *nd-DT*, and *DT*.

On behalf of this, we fix the set $M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}$ of tree transformation classes. Then we consider the monoid $[M]$ of all tree transformation classes of the form $X_1 \circ \ldots \circ X_m$, where $m \geq 0$ and the X_i's are elements of M. For arbitrary composition classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of the above form, we want to know whether some inclusion, equality, or incomparability holds between them. Clearly, it is enough if we can decide whether $\mathcal{C}_1 \subseteq \mathcal{C}_2$ holds.

As the main result of the chapter, we give an effective description of the monoid $[M]$ with respect to the inclusion. This means that we present an algorithm, which can decide, given arbitrary two elements of the monoid, whether some inclusion holds between them.

The main steps of the development of this algorithm are as follows:

1. We consider the free monoid M^* of strings generated by M. Then a unique homomorphism $|\ | : M^* \to [M]$ exists such that, for any $X_1, \ldots, X_n \in M$, $|X_1 \cdot \ldots \cdot X_n| = X_1 \circ \ldots \circ X_n$ (see [BurSan]). We denote the kernel of $|\ |$ by θ, that is, for any strings $u, v \in M^*$, $u\theta v$ if and only if $|u| = |v|$.
2. We present a confluent and terminating rewriting system $R \subseteq M^* \times M^*$ such that $\Leftrightarrow_R^* = \theta$, where $\Leftrightarrow_R^*$ is the reflexive, symmetric, and transitive closure of the reduction relation $\Rightarrow_R$ over M^*.
3. We present the inclusion diagram, i.e. the Hasse diagram with respect to the inclusion of the set $|NF(R)| = \{|u| \mid u \in NF(R)\}$, where $NF(R)$ is the set of R-normal forms.

The algorithm works as follows. Given two arbitrary composition classes $\mathcal{C} = X_1 \circ \ldots \circ X_n$ and $\mathcal{D} = Y_1 \circ \ldots \circ Y_m$, we form the strings $x = X_1 \cdot \ldots \cdot X_n$ and $y = Y_1 \cdot \ldots \cdot Y_m$, and compute the corresponding R-normal forms $x \Rightarrow_R^* u$ and $y \Rightarrow_R^* v$. Since R is terminating and confluent, u and v exist and unique. Moreover, $|x| = |u|$ and $|y| = |v|$ hold by $\Leftrightarrow_R^* = \theta$. Hence $\mathcal{C} \subseteq \mathcal{D}$ if and only if $|u| \subseteq |v|$. However, this latter can be decided by direct inspection of the inclusion diagram of $NF(R)$.

Finally, we summarize our results and mention some open problems regarding the superlinear property.

We note that the research has also resulted some by-products, which could be interesting for themself as well. These are the minimalization algorithm of deterministic top-down tree recognizers (see Subsection 1.4.4), the definition of semi-universal deterministic top-down tree recognizers and that the minimalization preserves the semi-universal property (see Section 2.2), and the decomposition equation $DT = op\text{-}ni\text{-}DT \circ nr\text{-}l\text{-}nd\text{-}HOM$ (see Lemma 2.1.9).

This thesis is strongly based on the papers [DánFül1], [DánFül2] and [Dán]. All results presented here appear in the above works. We shall refer to the corresponding paper at the beginning of each chapter.

Acknowledgement. I am grateful to Zoltán Fülöp for his expert guidance and valuable suggestions during the course of this research.

I also wish to express my thanks to Pál Gyenizse and László Bernátsky for their very useful comments on a pre-release version of this thesis.

Chapter 1

Preliminaries

In this chapter we introduce the notions and the notation, which are necessary for understanding.

Moreover, we recall some preliminary results used in later chapters. However, some of that results are referred to in a modified form (especially in Subsection 1.4.4), hence we present a proof, where it is necessary.

We note that the symbols i, j, k, l, m, and n denotes integers in the sequel.

1.1 Sets and relations

An *alphabet* is a finite and nonempty set.

For arbitrary sets A and B, we denote by $A \subseteq B$ that A is a subset of B, by $A \subset B$ that A is a proper subset of B, and by $A \bowtie B$ that neither $A \subseteq B$ nor $B \subseteq A$ hold.

We write $\mathrm{pow}(A)$ and $|A|$ for the power set and the cardinality of A, respectively. Moreover, $A \times B$ denotes the Cartesian product of A and B.

A subset X of $\mathrm{pow}(A)$ is called a *partition of* A if $\emptyset \notin X$, $\bigcup_{P \in X} P = A$, and, for any $P_1, P_2 \in X$ such that $P_1 \neq P_2$, $P_1 \cap P_2 = \emptyset$ hold. In this case the elements of X are also called *classes of the partition* X.

Given two sets A and B, an arbitrary subset θ of $A \times B$ is called a *relation* from A to B. We also write $a\theta b$ meaning that $(a, b) \in \theta$.

The *inverse relation of* θ is a relation θ^{-1} from B to A, where, for any $a \in A$ and $b \in B$, $b\theta^{-1}a$ holds if and only if $a\theta b$.

For a subset $A' \subseteq A$, we define the *restriction of* θ *to* A' as $\theta|_{A'} = \{(a, b) \in \theta \mid a \in A'\}$.

A relation from A to A is called also a *relation over* A. The *identity relation over* A is $\mathrm{id}(A) = \{(a, a) \mid a \in A\}$. A relation θ over a set A is called

- *reflexive* if, for each $a \in A$, $a\theta a$ holds,
- *transitive* if, for any $a, b, c \in A$, $a\theta b$ and $b\theta c$ imply $a\theta c$,

- *symmetric* if, for any $a, b \in A$, $a\theta b$ implies $b\theta a$,
- *equivalence* if it is reflexive, transitive, and symmetric,
- *antisymmetric* if, for any $a, b \in A$, $a\theta b$ and $b\theta a$ imply that a and b are identical, and
- *partial order* if it is reflexive, transitive, and antisymmetric.

An equivalence relation $\equiv$ over A defines a partition of A, where, for any $a, b \in A$, a and b are in the same class if and only if $a \equiv b$. We denote by $[a]_{\equiv}$ the class of an element $a \in A$ with respect to $\equiv$.

A pair $(A, \leq)$ is called *partially ordered set* if A is a set and $\leq$ is a partial order relation over A. A partially ordered set can be represented by a *Hasse diagram* (see [BurSan]).

Let H be a set of which the elements are classes and let $\subseteq$ be the inclusion relation over H. Clearly, $(H, \subseteq)$ is a partially ordered set. By the *inclusion diagram* of H we mean the Hasse diagram of $(H, \subseteq)$.

Let θ be a relation from A to B. For any $a \in A$, we put $\theta(a) = \{b \in B \mid a\theta b\}$. The sets $\text{dom}(\theta) = \{a \in A \mid \text{there is a } b \in B \text{ such that } a\theta b\}$ and $\text{range}(\theta) = \{b \in B \mid \text{there is an } a \in A \text{ such that } a\theta b\}$ are called the *domain* and the *range* of θ, respectively. We say that θ is *total* if $\text{dom}(\theta) = A$.

Let $\theta \subseteq A \times B$ and $\mu \subseteq B \times C$. The *composition* $\theta \circ \mu$ of the relations θ and μ is a relation from A to C defined as $\theta \circ \mu = \{(a, c) \mid \text{there is a } b \in B \text{ such that } a\theta b \text{ and } b\mu c\}$.

Let θ be a relation over A. The n-fold composition θ^n of θ with $n \geq 0$ is defined by $\theta^0 = \text{id}(A)$ and $\theta^i = \theta \circ \theta^{i-1}$, where $i > 0$. The *transitive closure* and the *reflexive and transitive closure* of θ are the relations $\theta^+ = \bigcup_{n\geq 1} \theta^n$ and $\theta^* = \bigcup_{n\geq 0} \theta^n$, respectively. Moreover, the *symmetric closure* of θ is $\theta \cup \theta^{-1}$.

We extend the concepts of domain, range, and composition for classes of relations. Let $\mathcal{C}$ and $\mathcal{C}'$ be classes of relations. The domain and the range of $\mathcal{C}$ are defined by $\text{dom}(\mathcal{C}) = \{\text{dom}(\theta) \mid \theta \in \mathcal{C}\}$ and $\text{range}(\mathcal{C}) = \{\text{range}(\theta) \mid \theta \in \mathcal{C}\}$, respectively. The composition of $\mathcal{C}$ and $\mathcal{C}'$ is the relation class $\mathcal{C} \circ \mathcal{C}' = \{\theta \circ \mu \mid \theta \in \mathcal{C} \text{ and } \mu \in \mathcal{C}'\}$.

For an arbitrary class $\mathcal{C}$ of relations, we also define the n-fold composition $\mathcal{C}^n$ of $\mathcal{C}$ with $n \geq 1$ as $\mathcal{C}^1 = \mathcal{C}$ and $\mathcal{C}^i = \mathcal{C} \circ \mathcal{C}^{i-1}$, where $i > 1$. The *transitive closure* of $\mathcal{C}$ under the composition is the relation class $\mathcal{C}^+ = \bigcup_{n\geq 1} \mathcal{C}^n$.

A class $\mathcal{C}$ of relations is said to be *closed under the composition* if $\mathcal{C}^2 \subseteq \mathcal{C}$ holds.

Let A and B be sets. A relation $\nu \subseteq A \times B$ is called a *mapping* from A to B, denoted by $\nu : A \to B$, if, for any $a \in A$, exactly one $b \in B$ exists such that $(a, b) \in \nu$ holds. In this case we also write $\nu(a) = b$ for $(a, b) \in \nu$.

A mapping ν from A to B is said to be

- *injective* if, for each $b \in B$, there is at most one $a \in A$ such that $\nu(a) = b$,

- *surjective* if range(ν) $= B$, and
- *bijective* if it is injective and surjective.

A bijective mapping is also called a *bijection*.

Finally, we introduce the concept of *hierarchy*. A family of classes $\{\mathcal{C}_k \mid k \geq 1\}$ is called a hierarchy if $\mathcal{C}_k \subseteq \mathcal{C}_{k+1}$ holds for each $k \geq 1$. Moreover, it is said to be *proper* if each inclusion is proper, i.e., for every $k \geq 1$, $\mathcal{C}_k \subset \mathcal{C}_{k+1}$ holds.

1.2 Strings and string rewriting systems

Let A be an alphabet. A *string* w over A is a finite sequence $a_1 \ldots a_l$, where $l \geq 0$ and $a_1, \ldots, a_l \in A$. Then the *length* of w, denoted by length(w), is l. The *empty string*, i.e. the sequence containing no letter, is denoted by e. Thus length(e) $= 0$. The set of all strings over A is denoted by A^*.

Let $w = a_1 \ldots a_l$ and $w' = b_1 \ldots b_k$ be strings over A. We define the *concatenation* $w_1 w_2$ of w_1 and w_2 as the string $a_1 \ldots a_l b_1 \ldots b_k$ over A. If we want to refer to the concatenation explicitly as an operation over A^*, then we shall denote it by $\cdot$.

We note that A^* is the free monoid generated by A with the concatenation operation, where e is the identity (see [BurSan]).

An equivalence relation $\equiv$ over A^* is called a *congruence* over A^* if, for any words $u_1, u_2, v_1, v_2 \in A^*$, $u_1 \equiv u_2$ and $v_1 \equiv v_2$ imply $u_1 v_1 \equiv u_2 v_2$.

A *string rewriting system* R over an alphabet A is a finite relation over A^*. The elements of R are called *rewriting rules* and we write $u \to v$ for $(u, v) \in R$.

The *reduction relation* $\Rightarrow_R$ over A^* induced by R is defined as follows. For any strings $w, w' \in A^*$, $w \Rightarrow_R w'$ holds if and only if a rule $v \to v' \in R$ and strings $u_1, u_2 \in A^*$ exist such that $w = u_1 v u_2$ and $w' = u_1 v' u_2$. We write $\Leftarrow_R$ for $\Rightarrow_R^{-1}$.

The reflexive, transitive, and symmetric closure of $\Rightarrow_R$, denoted by $\Leftrightarrow_R^*$, is a congruence over A^*. Informally speaking, $w \Leftrightarrow_R^* w'$ holds if and only if there is a sequence $w_0, \ldots, w_n$ of strings, for some $n \geq 0$, such that $w_0 = w$, $w_n = w'$ and, for every $1 \leq i \leq n$, either $w_{i-1} \Rightarrow_R w_i$ or $w_i \Rightarrow_R w_{i-1}$ holds.

We say that a string rewriting system R is

- *terminating* if there is no infinite sequence of the form $w_1 \Rightarrow_R w_2 \Rightarrow_R \ldots$ and
- *confluent* if, for any $v, w, w' \in A^*$, $v \Rightarrow_R^* w$ and $v \Rightarrow_R^* w'$ imply that a string $v' \in A^*$ exists such that $w \Rightarrow_R^* v'$ and $w' \Rightarrow_R^* v'$ hold.

A string $w \in A^*$ is called an *R-normal form* (or simply *normal form* if R is understood) if there is no $w' \in A^*$ such that $w \Rightarrow_R w'$. The set of R-normal forms is denoted by $NF(R)$. A string w' is called a normal form of a string w with respect to R if $w \Rightarrow_R^* w'$ and w' is an R-normal form.

We recall the following result (see [Boo]):

Propositon 1.2.1 *A terminating string rewriting system R is confluent if and only if each word of A^* has exactly one normal form.*

Observe that the above statement can also be explained as follows. Considering the partition of A^* with respect to the congruence $\Leftrightarrow_R^*$, a terminating R is confluent if and only if each $\Leftrightarrow_R^*$-class contains exactly one normal form.

We now mention a sufficient condition for a string rewriting system R to be terminating.

A *weight function* is a mapping $\rho : A \to \{1, 2, \ldots\}$, where $\rho(a)$ is called the weight of $a \in A$. Then ρ can be extended to a mapping $\rho : A^* \to \{1, 2, \ldots\}$ by letting $\rho(e) = 0$ and, for every $w \in A^*$ and $a \in A$, $\rho(wa) = \rho(w) + \rho(a)$.

We say that R is *weight reducing* if a weight function ρ exists such that, for each rule $u \to v$, $\rho(u) > \rho(v)$ holds. It should be clear that a weight reducing string rewriting system is necessarily terminating.

For more information about string rewriting systems we refer the reader to [Boo], [BooOtt], and [Huet].

1.3 Trees, tree languages, and tree transformations

A *ranked alphabet* Σ is an alphabet, in which every symbol has a unique rank in the set of nonnegative integers. For each $m \geq 0$, the set of symbols in Σ having rank m is denoted by Σ_m. We write $\Sigma = \{\sigma_1^{(m_1)}, \ldots, \sigma_n^{(m_n)}\}$ meaning that $\Sigma = \{\sigma_1 \ldots, \sigma_n\}$ is a ranked alphabet, where the symbol σ_i has the rank m_i, for each $1 \leq i \leq n$.

Let Σ be a ranked alphabet. For a set H, the *set of trees over Σ indexed by H* is denoted by $T_\Sigma(H)$ and it is defined as the smallest set U satisfying the following two conditions:

(i) $H \cup \Sigma_0 \subseteq U$.

(ii) $\sigma(t_1, \ldots, t_m) \in U$, whenever $m > 0$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in U$.

The set $T_\Sigma(\emptyset)$ of *ground trees* over Σ is written as T_Σ.

We specify a countable set $X = \{x_1, x_2, \ldots\}$ of symbols, called *variables*, and we put $X_m = \{x_1, \ldots, x_m\}$, for every $m \geq 0$. We assume that X is disjoint to any ranked alphabet. We write $T_{\Sigma,m}$ for $T_\Sigma(X_m)$.

Trees can be represented as expressions with parentheses. For instance, if $\Sigma = \{\delta^{(2)}, \sigma^{(1)}, \#^{(0)}\}$ then $\delta(\sigma(\#), \#) \in T_\Sigma$ and $\delta(\delta(x_1, \#), \delta(\sigma(x_2), \sigma(x_1))) \in T_{\Sigma,2}$. Moreover, these trees can be depicted as in Figure 1.1.

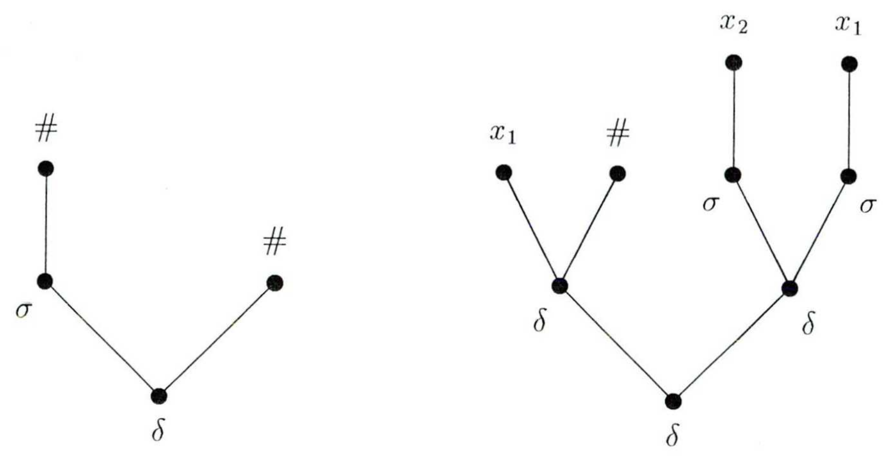

Figure 1.1: Representation of trees

A "chain" tree, like $\sigma(\ldots\sigma(\#)\ldots)$, where $\sigma \in \Sigma_1$ occurs i times, is abbreviated by $\sigma^i(\#)$. For example, $\sigma^3(\#)$ denotes the tree $\sigma(\sigma(\sigma(\#)))$.

We distinguish a subset $\hat{T}_{\Sigma,m}$ of $T_{\Sigma,m}$ as follows. A tree $t \in T_{\Sigma,m}$ is in $\hat{T}_{\Sigma,m}$ if and only if each variable in X_m appears exactly once in t and the order of the variables from left to right in t is exactly $x_1, \ldots, x_m$. For instance, if $\Sigma = \{\sigma^{(2)}\}$, then $\sigma(x_1, x_2) \in \hat{T}_{\Sigma,2}$, but $\sigma(x_1, x_1), \sigma(x_2, x_1) \notin \hat{T}_{\Sigma,2}$.

Let $t \in T_{\Sigma,m}$, for some $m \geq 0$. We define the *height of* t, the *set of subtrees of* t, and the *set of variables occurring in* t, denoted by height(t), sub(t), and var(t), respectively, as follows.

(*i*) If $t = x_i \in X_m$, then $\mathrm{height}(t) = 0$, $\mathrm{sub}(t) = \{x_i\}$, and $\mathrm{var}(t) = \{x_i\}$.

(*ii*) If $t = \sigma \in \Sigma_0$, then $\mathrm{height}(t) = 0$, $\mathrm{sub}(t) = \{\sigma\}$, and $\mathrm{var}(t) = \emptyset$.

(*iii*) If $t = \sigma(t_1, \ldots, t_n)$, where $n > 0$, $\sigma \in \Sigma_n$ and $t_1, \ldots, t_n \in T_{\Sigma,m}$, then $\mathrm{height}(t) = 1 + \max\{\mathrm{height}(t_i) \mid 1 \leq i \leq n\}$, $\mathrm{sub}(t) = \{t\} \cup \bigcup_{1 \leq i \leq n} \mathrm{sub}(t_i)$, and $\mathrm{var}(t) = \bigcup_{1 \leq i \leq n} \mathrm{var}(t_i)$.

We note that if $t = \sigma(t_1, \ldots, t_n)$, where $n > 0$, then, for any $1 \leq i \leq n$, t_i is called the *ith direct subtree* of t.

We introduce the concept of *tree substitution*. Let $m \geq 0$, $t \in T_{\Sigma,m}$, and $s_1, \ldots, s_m \in S$ where S is an arbitrary set of trees. We denote by $t[s_1, \ldots, s_m]$ the tree, which is obtained from t by replacing each occurrence of x_i in t by s_i, for every $1 \leq i \leq m$. Clearly, $t[s_1, \ldots, s_m] \in T_\Sigma(S)$ holds.

Let Σ be a ranked alphabet. A *tree language* L over Σ is an arbitrary subset $L \subseteq T_\Sigma$.

Let $\sigma \in \Sigma_m$, where $m \geq 1$, and let $L_1, \ldots, L_m$ be tree languages over Σ. The expression $\sigma(L_1, \ldots, L_m)$ denotes the tree language $\{\sigma(t_1, \ldots, t_m) \mid t_1 \in L_1, \ldots, t_m \in L_m\}$ over Σ.

Let Σ and Δ be ranked alphabets. A *tree transformation from* T_Σ *to* T_Δ is a relation from T_Σ to T_Δ. Since tree transformations are relations, the concepts of their domain, range and composition should be clear. Note that if $\tau \subseteq T_\Sigma \times T_\Delta$, then $\mathrm{dom}(\tau)$ and $\mathrm{range}(\tau)$ are tree languages over Σ and Δ, respectively.

We specify the class $I = \{\mathrm{id}(T_\Sigma) \mid \Sigma \text{ is a ranked alphabet}\}$ of identity tree transformations. Observe that if $I \subseteq \mathcal{C}$, then $\mathcal{C}^2 \subseteq \mathcal{C}$ holds (i.e. $\mathcal{C}$ is closed under the composition) if and only if $\mathcal{C}^2 = \mathcal{C}$.

Let $\mathcal{C}$ be a class of tree transformations. Now it is possible to define the *reflexive and transitive closure* of $\mathcal{C}$ as $\mathcal{C}^* = \bigcup_{n \geq 0} \mathcal{C}^n$, where $\mathcal{C}^0 = I$.

1.4 Top-down tree transducers

A *top-down tree transducer* is a 5-tuple $T = (Q, \Sigma, \Delta, q_0, R)$, where

- Q is an unary ranked alphabet, meaning that $Q = Q_1$, called the set of *states*, such that $Q \cap (\Sigma \cup \Delta) = \emptyset$.

- Σ and Δ are ranked alphabets, called the *input* and the *output* ranked alphabet, respectively.
- $q_0 \in Q$ is a distinguished element of Q, called the *initial state*.
- R is a finite set of rules of the form

$$q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})], \qquad (*)$$

  where $m, n \geq 0$, $\sigma \in \Sigma_m$, $1 \leq i_1, \ldots, i_n \leq m$, $q, q_1, \ldots, q_n \in Q$, and $t \in \hat{T}_{\Delta,n}$.

A rule as above will be referred to as a q-rule for σ or a (q, σ)-rule for brevity.

A rule of the form $(*)$ is said to be *reducing* if $t = x_1$ holds, i.e. it is of the form $q(\sigma(x_1, \ldots, x_m)) \to q'(x_i)$, for some $q' \in Q$ and $1 \leq i \leq m$.

The rules in R induce a relation, called *derivation* and denoted by $\Rightarrow_T$, over the set $T_\Delta(Q(T_\Sigma))$, where $Q(T_\Sigma)$ denotes the set $\{q(t) \mid q \in Q, t \in T_\Sigma\}$. It is defined as follows.

For any trees $r, s \in T_\Delta(Q(T_\Sigma))$, $r \Rightarrow_T s$ holds if and only if there is a rule $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R such that s can be obtained from r by replacing an occurrence of a subtree $q(\sigma(t_1, \ldots, t_m))$ of r by

$$t[q_1(t_{i_1}), \ldots, q_n(t_{i_n})],$$

where $t_1, \ldots, t_m \in T_\Sigma$.

The *tree transformation* τ_T *induced by* T is defined as

$$\tau_T = \{(r, s) \in T_\Sigma \times T_\Delta \mid q_0(r) \Rightarrow_T^* s\}.$$

We note that top-down tree transducers are sometimes defined to have more than one initial states. However, that concept is not essentially different from our one. It is an easy exercise to show that, for each top-down tree transducer having more initial states, a top-down tree transducer with one initial state can be constructed, which induces the same tree transformation.

We say that T is *deterministic* if, for any $q \in Q$ and $\sigma \in \Sigma$, there is at most one (q, σ)-rule in R. The expression deterministic top-down tree transducer is abbreviated to dt tree transducer in the sequel.

A tree transformation τ is called a *dt tree transformation* if a dt tree transducer T exists so that $\tau = \tau_T$. The class of all dt tree transformations is denoted by DT.

We suppose that T is deterministic in what follows.

Consider an arbitrary (q, σ)-rule in R of the above $(*)$ form. The term $t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ is called the *right-hand side* of the rule and it is denoted by $\mathrm{rhs}(q, \sigma)$.

We note that the order of the variables from left to right occurring in the above right-hand side is $x_{i_1}, \ldots, x_{i_n}$, because the order of the ones from left to right occurring in t is $x_1, \ldots, x_n$.

Again, consider an arbitrary (q, σ)-rule in R of the $(*)$ form. For each $1 \leq j \leq m$, we define $\mathrm{rst}(q, \sigma, j) = \{q_k \in Q \mid 1 \leq k \leq n, i_k = j\}$, i.e. the set of states applied to x_j in $\mathrm{rhs}(q, \sigma)$.

1.4.1 Restricted types

We introduce some restricted subtypes of dt tree transducers applying certain restrictions to the form of rules. Moreover, we specify a unique abbreviation for the name of each type. Finally, we define a way to construct combined types using the ones listed bellow.

We note that the types (1), (2), (4), and (8) are well known from the theory of tree transducers (see, e.g., [GécSte4]). Moreover, (5), (6), and (7) were also defined in [AndBos].

Let $T = (Q, \Sigma, \Delta, q_0, R)$ be a dt tree transducer. We say that T is:

(1) *Total* (t) if, for any $\sigma \in \Sigma$ and $q \in Q$, there exists a (q, σ)-rule in R. Note that, being T deterministic, this implies that there is exactly one (q, σ)-rule in R, for any $q \in Q$ and $\sigma \in \Sigma$. Clearly, in this case τ_T is a total tree transformation.

(2) *Linear* (l) if, for every rule $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, each of the variables $x_1, \ldots, x_m$ appears at most once in the right-hand side. Note that in this case $m \geq n$.

(3) *Superlinear* (sl) if it is linear and, for every $\sigma \in \Sigma_m$ with $m \geq 0$ and any two different states $q, q' \in Q$, $\mathrm{var}(\mathrm{rhs}(q, \sigma)) \cap \mathrm{var}(\mathrm{rhs}(q', \sigma)) = \emptyset$ holds. Equivalently, T is sl-dt, if it is linear and, for every $\sigma \in \Sigma_m$ with $m \geq 0$ and $1 \leq i \leq m$, there is at most one state $q \in Q$ such that x_i occurs in $\mathrm{rhs}(q, \sigma)$.

(4) *Nondeleting* (nd) if, for every $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, each of the variables $x_1, \ldots, x_m$ appears at least once in the right-hand side. Note that in this case $m \leq n$.

(5) *Order preserving* (op) if, for every $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, the order $i_1 \leq \ldots \leq i_n$ holds.

(6) *Nonreducing* (nr) if R does not contain reducing rule, i.e. there is no rule of the form $q(\sigma(x_1, \ldots, x_m)) \to q'(x_i)$, where $1 \leq i \leq m$ in R.

(7) *Nonincreasing* (ni) if, for every $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, it holds that $\mathrm{height}(t) \leq 1$. Note that in this case either $t = x_1$ or $t = \delta(x_1, \ldots, x_n)$, for some $\delta \in \Delta_n$.

(8) *Relabeling (rl)*, if each rule in R is of the form

$$q(\sigma(x_1, \ldots, x_m)) \to \delta(q_1(x_1), \ldots, q_m(x_m)),$$

where $m \geq 0, \sigma \in \Sigma_m, \delta \in \Delta_m$. Roughly speaking, processing a tree, T does not change the skeleton, only relabels the nodes.

(9) *Homomorphism* (hom) if it is total and Q is a singleton set, i.e. $Q = \{q_0\}$.

These attributes can be combined. For instance, by an op-l-nd-dt tree transducer, we mean an order preserving, linear and nondeleting deterministic top-down tree transducer.

Let x be an arbitrary combination of some of the modifiers in {t, l, sl, nd, op, nr, ni, rl, hom} such as l-nd, op-ni-sl, etc. A dt tree transformation is said to be an x-dt tree transformation if it can be induced by an x-dt tree transducer.

Observe that the relabeling deterministic top-down tree transducers are exactly the nonincreasing, nonreducing, linear, and nondeleting deterministic top-down tree transducers, hence rl is in fact a shorthand for the combination ni-nr-l-nd.

The class of all x-dt tree transformations is denoted by x-DT. Clearly, the order of the modifiers in a combination is irrelevant from the point of view of meaning, i.e. if x is a combination of modifiers and y is a permutation of x then x-DT = y-DT holds. Moreover, we can assume without loss of generality that any modifier occurs at most once in a combination.

We write simply hom instead of hom-dt. For example, op-l-nd-hom means an order preserving, linear and nondeleting homomorphism dt tree transducer.

Let x be a combination of modifiers as above. The class of all x-hom tree transformations is denoted by x-HOM.

We note that, for any combination x, both $I \subseteq x\text{-}DT$ and $I \subseteq x\text{-}HOM$ hold. Observe that if $\mathcal{C}$ and $\mathcal{D}$ are tree transformation classes and $I \subseteq \mathcal{D}$, then $\mathcal{C} \subseteq \mathcal{C} \circ \mathcal{D}$. This follows from the fact that every tree transformation τ in $\mathcal{C}$ can be decomposed as $\tau = \tau \circ \iota$, where ι is a suitable identity in $\mathcal{D}$. Specially, $x\text{-}DT^n \subseteq x\text{-}DT^{n+1}$ and $x\text{-}HOM^n \subseteq x\text{-}HOM^{n+1}$ hold for every $n \geq 0$.

1.4.2 Compositions and decompositions

In this subsection we introduce the concept of syntactic composition of dt tree transducers. Moreover, we clarify the correspondence between compositions of dt tree transformations and syntactic compositions of dt tree transducers. Finally, we recall some earlier results concerning compositions of dt tree transformation classes.

Let $T = (Q, \Sigma, \Delta, q_0, R)$ and $T' = (Q', \Delta, \Omega, q_0', R')$ be dt tree transducers. By the *syntactic composition of T and T'* we mean the dt tree transducer

$$T \circ T' = (Q' \times Q, \Sigma, \Omega, (q_0', q_0), R''),$$

where R'' is the smallest set of rules, which satisfies the following condition. If

(1) there is a rule $q(\sigma(x_1,\ldots,x_m)) \to t[q_1(x_{i_1}),\ldots,q_n(x_{i_n})] \in R$ and

(2) there is a state $p \in Q'$ such that $p(t) \Rightarrow^*_{T'} t'[p_1(x_{j_1}),\ldots,p_k(x_{j_k})]$, for some $t' \in \hat{T}_{\Omega,k}$ with $k \geq 0$ and $p_1,\ldots,p_k \in Q'$,

then the rule

$$(p,q)(\sigma(x_1,\ldots,x_m)) \to t'[(p_1,q_{j_1})(x_{i_{j_1}}),\ldots,(p_k,q_{j_k})(x_{i_{j_k}})]$$

is in R''. It is easy to see that R'' is finite. Roughly speaking, the composition transducer works such that

(1) applies the appropriate (q,σ)-rule of T and

(2) lets T' process the tree t, obtained by the application of the above rule, starting with state p.

Reducing rules play a very important role in the characterization of the syntactic composition of sl-dt tree transducers, hence we pay more attention for them.

Suppose that there is a reducing rule in R, i.e. a rule of the form

$$q(\sigma(x_1,\ldots,x_m)) \to q'(x_i),$$

where $m \geq 1$ and $1 \leq i \leq m$. Clearly, for every $p \in Q'$, $p(x_1) \Rightarrow^*_{T'} p(x_1)$ holds. Therefore the rule

$$(p,q)(\sigma(x_1,\ldots,x_m)) \to (p,q')(x_i)$$

is in R''. Informally speaking, a reducing rule for σ in R as above induces $|Q'|$ different reducing rules for σ in R'' such that each of them contains x_i in the right-hand side.

Now we show the correspondence between syntactic compositions of dt tree transducers and compositions of dt tree transformations. The following results implicitly appear in [Bak3] and [Eng1], but in the present form they are stated in [FülVág1].

Propositon 1.4.1 *For any dt tree transducers T and T', the following statements hold:*

(1) If T is total or T' is nondeleting, then $\tau_{T\circ T'} = \tau_T \circ \tau_{T'}$.

(2) $\tau_{T\circ T'}|_{dom(\tau_T)} = \tau_T \circ \tau_{T'}$.

(3) Let x be any of the modifiers $\{t, l, nd, hom\}$. If both T and T' are x-dt tree transducers, then $T \circ T'$ is also x-dt.

We do not present the formal proofs of the above statements. However, it is worth to recall informally the main points of them.

To prove (1) and (2), it is enough to see that $T \circ T'$ processes every tree on almost the same way, as the consecutive application of T and T' does. The only difference is the following.

Suppose that τ_T is partial and T' is not nondeleting. Then there is a tree t on which T fails, i.e. $t \notin \mathrm{dom}(\tau_T)$ and hence $t \notin \mathrm{dom}(\tau_T \circ \tau_{T'})$. However, since T' is not nondeleting, it can occur that, informally speaking, the rules of the dt tree transducer $T \circ T'$ deletes the subtrees of t, on which T fails, without processing. Thus $t \in \mathrm{dom}(\tau_{T \circ T'})$ can hold.

Clearly, this implies $\tau_T \circ \tau_{T'} \subseteq \tau_{T \circ T'}$ and that the converse inclusion generally does not hold.

As for (3), it is straightforward to show, by the construction of the rules of $T \circ T''$.

A large series of equations concerning compositions of dt tree transformation classes can be derived from Proposition 1.4.1. We present a short list of them containing that ones, which are necessary in our proofs.

Corollary 1.4.2

(1)	$t\text{-}DT \circ DT$	$=$	DT
(2)	$DT \circ nd\text{-}DT$	$=$	DT
(3)	$nd\text{-}HOM \circ nd\text{-}HOM$	$=$	$nd\text{-}HOM$
(4)	$nd\text{-}HOM \circ DT$	$=$	DT
(5)	$DT \circ nd\text{-}HOM$	$=$	DT
(6)	$nr\text{-}l\text{-}nd\text{-}HOM \circ DT$	$=$	DT
(7)	$nd\text{-}HOM \circ nd\text{-}DT$	$=$	$nd\text{-}DT$

Characterizing a class of dt tree transformations, it is always a pivot question whether it is closed under the composition. By Proposition 1.4.1, it is easy to show that the classes $t\text{-}DT$ and $nd\text{-}DT$ are.

On the other hand, it has turned out that the tree transformation classes DT and $l\text{-}DT$ are not closed under the composition, i.e. $DT \subset DT^2$ and $l\text{-}DT \subset l\text{-}DT^2$ (see [Rou] or [GécSte4]).

In this case the question naturally arises whether the powers of these classes constitute proper hierarchies. The following proposition, cited from [FülVág1], shows that they do not.

Propositon 1.4.3 *For any $n \geq 2$, it holds that $DT^n = DT^2$ and $l\text{-}DT^n = l\text{-}DT^2$.*

Finally, we recall a decomposition equation from [Bak3], which shows that every dt tree transformation can be induced by consecutive application of an nd-hom and an l-dt tree transducer. Note that this result also appears in [GécSte4].

Propositon 1.4.4 $DT = nd\text{-}HOM \circ l\text{-}DT$

1.4.3 Top-down tree recognizers and recognizable tree languages

A *top-down tree recognizer (ttr)* $T = (Q, \Sigma, \Sigma, q_0, R)$ is a top-down tree transducer, of which the rules are of the form

$$q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)),$$

where $m \geq 0$, $\sigma \in \Sigma_m$, and $q, q_1, \ldots, q_m \in Q$. If T is deterministic, then it is called a *deterministic top-down tree recognizer (dttr)*.

Observe that $\tau_T \subseteq \mathrm{id}(T_\Sigma)$ holds, i.e. τ_T is a partial identity over T_Σ. Moreover, if T is deterministic, then it is an rl-dt tree transducer.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. We say that a state $q \in Q$ is *universal*, if, for all $t \in T_\Sigma$, $q(t) \Rightarrow_T^* t$ holds, i.e. $\{t \in T_\Sigma \mid q(t) \Rightarrow_T^* t\} = T_\Sigma$. Observe that, for any rule $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, if q is universal, then $q_1, \ldots, q_m$ are necessarily universal, too.

We say that T *recognizes* the tree $t \in T_\Sigma$ if $q_0(t) \Rightarrow_T^* t$. The *tree language recognized by* T is $L(T) = \{t \in T_\Sigma \mid q_0(t) \Rightarrow_T^* t\}$. Observe that $L(T) = \mathrm{dom}(\tau_T)$.

A tree language is *recognizable* (resp. *deterministic recognizable*) if it is recognized by a ttr (resp. dttr). We denote by REC (resp. $DREC$) the class of recognizable (resp. deterministic recognizable) tree languages.

Note that the original concept of recognizability concerning tree languages is defined by descending (or bottom-up) tree automata, see in [GécSte4]. However, consulting Chapter II in [GécSte4], one can easily see that top-down tree recognizers are equivalent to regular tree grammars in normal form and hence to descending tree automata.

Clearly, $DREC \subseteq REC$ holds. Moreover, it is a well-known result (see, e.g., [GécSte4]) that the inclusion is proper, i.e. $DREC \subset REC$.

1.4.4 Minimal deterministic top-down tree recognizers

Deterministic top-down tree recognizers also have automaton type equivalent, namely *deterministic ascending (or top-down) tree automata (dtta)*. A short reflection will show that there are mainly notational differences between these types of devices, hence we can apply the notions and results in [GécSte3] to dttr's without difficulties.

An n-ary dtta is a 5-tuple $A = (Q, \Sigma, Y_n, q_0, F)$, where $n > 0$,

- Q is the finite nonempty set of states,
- $q_0 \in Q$ is the initial state,
- $F = (Q_1, \ldots, Q_n) \in (\mathrm{pow}(Q))^n$ is the final state vector,

- $Y_n = \{y_1, \ldots, y_n\}$ is the set of automaton variables, and
- Σ is a ranked alphabet, where $\Sigma \cap Y_n = \emptyset$, $\Sigma_0 = \emptyset$, and every $\sigma \in \Sigma_m$ with $m > 0$ is realized as a mapping $\sigma^A : Q \to Q^m$.

We now specify how a dtta A recognizes trees. Define the mapping $\alpha_A : T_\Sigma(Y_n) \to \text{pow}(Q)$ as follows.

(i) $\alpha_A(y_i) = Q_i$, for $1 \leq i \leq n$, and

(ii) $\alpha_A(t) = \{q \in Q \mid \sigma^A(q) \in \alpha_A(t_1) \times \ldots \times \alpha_A(t_m)\}$, if $t = \sigma(t_1, \ldots, t_m)$ with $m > 0$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in T_\Sigma(Y_n)$.

The tree language recognized by A is $L(A) = \{t \in T_\Sigma(Y_n) \mid q_0 \in \alpha_A(t)\}$.

We show that, for any dttr, an equivalent dtta can be constructed.

Construction 1.4.5 *Consider an arbitrary dttr $T = (Q, \Sigma, \Sigma, q_0, R)$ and suppose that $\Sigma_0 = \{\delta_1, \ldots, \delta_n\}$ with $n > 0$. Let $p \notin Q$ be a new state. Define the dtta $A = (Q \cup \{p\}, \Sigma - \Sigma_0, \Sigma_0, q_0, F)$, where*

- *$F = (Q_1, \ldots, Q_n)$ with $Q_i = \{q \in Q \mid q(\delta_i) \to \delta_i \in R\}$, for $1 \leq i \leq n$, and,*
- *for all $m > 0$, $\sigma \in \Sigma_m$, and $q \in Q \cup \{p\}$, if $q \in Q$ and $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ is in R, then let $\sigma^A(q) = (q_1, \ldots, q_m)$, otherwise let $\sigma^A(q) = (p, \ldots, p)$.*

It is straightforward to prove $L(A) = L(T)$. Note that the case $n = 0$ is trivial, because then $L(T) = \emptyset$.

Conversely, for any dtta, an equivalent dttr can be constructed.

Construction 1.4.6 *Let $A = (Q, \Sigma, Y_n, q_0, F)$ be an arbitrary dtta. Assign the rank 0 to each element of Y_n and let $\Delta = \Sigma \cup Y_n$. Define the dttr $T = (Q, \Delta, \Delta, q_0, R)$, where R is constructed as follows:*

(i) for all $1 \leq i \leq n$, $q(y_i) \to y_i \in R$ if and only if $q \in Q_i$ and,

(ii) for all $m > 0$, $\sigma \in \Sigma_m$ and $q, q_1, \ldots, q_m \in Q$, the rule $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ is in R if and only if $\sigma^A(q) = (q_1, \ldots, q_m)$.

It is easy to show $L(T) = L(A)$.

We now recall some definitions and results from [GécSte3] using the dttr notation. Note that two dttr's are called equivalent if they recognize the same tree language.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. A state $q \in Q$ of T is called *0-state* if the set $\{t \in T_\Sigma \mid q(t) \Rightarrow_T^* t\}$ is empty.

A dttr $T = (Q, \Sigma, \Sigma, q_0, R)$ is said to be *normalized*, if either it has no 0-state, or the only 0-state is q_0 and in this case $Q = \{q_0\}$ and $R = \emptyset$ hold.

We note that this concept of normalization differs from the original one in [GécSte3] on page 40. Namely, if a dttr is normalized by the above definition, then the dtta given by the Construction 1.4.5 is normalized in the sense of [GécSte3]. However, the converse is not true, that is if a dtta is normalized in the sense of [GécSte3], then the dttr given by the Construction 1.4.6 is generally not normalized by the above definition.

The difference follows from the fact that in the case of dtta's every $\sigma \in \Sigma_m$ with $m > 0$ is realized as a mapping $\sigma^A : Q \to Q^m$, hence σ^A is defined for each $q \in Q$. That is why the 0-states cannot be discarded completely from the state set of a normalized dtta. On the other hand, the 0-states (except q_0) and the corresponding rules can be deleted without difficulties in the case of dttr's, as it is shown by the following proposition.

Propositon 1.4.7 *For any dttr $T = (Q, \Sigma, \Sigma, q_0, R)$, an equivalent normalized dttr $T_{nor} = (Q', \Sigma, \Sigma, q_0, R')$ can be constructed effectively such that $Q' \subseteq Q$ and $R' \subseteq R$.*

Proof. The set of non 0-states can be computed as follows. Define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \ldots$ of subsets of Q, where

(i) $Q^{(0)} = \{q \in Q \mid \exists \delta \in \Sigma_0 : q(\delta) \to \delta \in R\}$ and,

(ii) for $i \geq 0$, $Q^{(i+1)} = Q^{(i)} \cup \{q \in Q \mid \exists m \geq 1, \sigma \in \Sigma_m : q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ and $q_1, \ldots, q_m \in Q^{(i)}\}$.

Obviously, there is a $k \geq 0$ such that $Q^{(k)} = Q^{(k+j)}$, for every $j \geq 1$.

If $q_0 \notin Q^{(k)}$, then let $Q' = \{q_0\}$ and $R' = \emptyset$. Clearly, in this case $L(T) = L(T_{nor}) = \emptyset$ holds.

Finally, if $q_0 \in Q^{(k)}$, then let $Q' = Q^{(k)}$ and $R' = \{q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R \mid q, q_1, \ldots, q_m \in Q^{(k)}\}$. Observe that, for any $t \in L(T)$, during the derivation $q_0(t) \Rightarrow_T^* t$ only such rules are applied, which do not contain a 0-state, hence $q_0(t) \Rightarrow_{T_{nor}}^* t$, too. Therefore $L(T_{nor}) = L(T)$. □

We define the binary relation $\mapsto_T$ over Q as follows. Let $q, p \in Q$, then $q \mapsto_T p$ if and only if there exists a $\sigma \in \Sigma_m$ with $m > 0$ such that p appears in $\mathrm{rhs}(q, \sigma)$. We say that p is *accessible* from q if $q \mapsto_T^* p$ holds. The dttr T is called *connected* if every state in Q is accessible from q_0.

Note that the above concept of accessibility is derived from the concept of reachability of states of dtta's defined in [GécSte3] on pages 41-42.

Propositon 1.4.8 *For any dttr $T = (Q, \Sigma, \Sigma, q_0, R)$, an equivalent connected dttr $T_{con} = (Q', \Sigma, \Delta, q_0, R')$ can be constructed effectively such that $Q' \subseteq Q$ and $R' \subseteq R$. Moreover, if T is normalized, then T_{con} is also normalized.*

Proof. The set of accessible states can be determined in the following way. Define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \dots$ of subsets of Q, where

(i) $Q^{(0)} = \{q_0\}$ and,

(ii) for $i \geq 0$, $Q^{(i+1)} = Q^{(i)} \cup \{q \in Q \mid \exists m \geq 1, \sigma \in \Sigma_m, p \in Q^{(i)} : q$ occurs in $\mathrm{rhs}(p, \sigma)\}$.

Clearly, there is a $k \geq 0$ such that $Q^{(k)} = Q^{(k+j)}$, for every $j \geq 1$.

Let $Q' = Q^{(k)}$ and $R' = \{q(\sigma(x_1, \dots, x_m)) \to \sigma(q_1(x_1), \dots, q_m(x_m)) \in R \mid q, q_1, \dots, q_m \in Q^{(k)}\}$. It is easy to show that $L(T_{con}) = L(T)$.

Note that the construction of Q' is derived from the procedure computing H_k's in [GécSte3] on page 43.

Moreover, by the construction of T_{con}, it should be obvious that if T is normalized, then T_{con} is also normalized. □

By Proposition 1.4.8, if T is a normalized dttr, then T_{con} is also normalized. However, the converse is not true, i.e. if T is a connected dttr, then T_{nor} is not necessarily connected. To see this, consider the following example.

Let $T = (\{q_0, q_1, q_2\}, \Sigma, \Sigma, q_0, R)$, where $\Sigma = \{\sigma^{(2)}, \#^{(0)}\}$ and $R = \{q_0(\#) \to \#,\ q_1(\#) \to \#,\ q_0(\sigma(x_1, x_2)) \to \sigma(q_1(x_1), q_2(x_2))\}$. Clearly, T is a connected dttr, but $T_{nor} = (\{q_0, q_1\}, \Sigma, \Sigma, q_0, \{q_0(\#) \to \#,\ q_1(\#) \to \#\})$ is not connected, because q_1 is not an accessible state in T_{nor}.

If we refer to the construction of the normalized and connected equivalent dttr $T_{nor,con}$ of a dttr T in the sequel, then we always mean that T_{nor} must be determined first from T as defined in the proof of Proposition 1.4.7, and $T_{nor,con}$ must be computed from T_{nor} as specified in the proof of Proposition 1.4.8.

A dttr T is said to be *minimal* if, for every dttr T' such that T' is equivalent to T, $|Q| \leq |Q'|$ holds, where Q and Q' are the sets of states of T and T', respectively (cf. minimal dtta on page 38 in [GécSte3]).

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ and $T' = (Q', \Sigma, \Sigma, q_0', R')$ be dttr's. We say that T and T' are *isomorphic* if there exists a bijection $\nu : Q \to Q'$ such that $\nu(q_0) = q_0'$ holds and, for any $m \geq 0$, $\sigma \in \Sigma_m$, and states $q, q_1, \dots, q_m \in Q$, the rule $q(\sigma(x_1, \dots, x_m)) \to \sigma(q_1(x_1), \dots, q_m(x_m))$ is in R if and only if the rule $\nu(q)(\sigma(x_1, \dots, x_m)) \to \sigma(\nu(q_1)(x_1), \dots, \nu(q_m)(x_m))$ is in R'. In this case ν is also called a *dttr isomorphism* (cf. dtta isomorphism on page 39 in [GécSte3]). Note that if T and T' are isomorphic, then clearly $|Q| = |Q'|$ and they are equivalent.

We say that a minimal dttr T is *unique up to isomorphism* if, for each minimal dttr T', which is equivalent to T, it holds that T' and T are isomorphic.

The following result is derived from Theorem 8 in [GécSte3].

Propositon 1.4.9 *For any dttr T, an equivalent minimal dttr T_{min} exists. Moreover, it is unique up to isomorphism and can effectively be constructed.*

We also present the construction of the minimal dttr. We note that the following construction is derived from the construction of reduced dtta presented in [GécSte3] on page 43.

Construction 1.4.10 *Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. By Propositions 1.4.7 and 1.4.8, we can assume without loss of generality that T is normalized and connected (if it is not, then consider $T_{nor,con}$ instead of T). We define a sequence $\equiv_0 \supseteq \equiv_1 \supseteq \ldots$ of equivalence relations over Q, where*

(i) $q \equiv_0 p$ if and only if, for every $\sigma \in \Sigma_0$, $q(\sigma) \to \sigma \in R$ holds if and only if $p(\sigma) \to \sigma \in R$ and,

(ii) for $i \geq 0$, $q \equiv_{i+1} p$ if and only if $q \equiv_i p$ and, for every $\sigma \in \Sigma_m$ with $m > 0$, either both q and p are not defined on σ, or both $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma(p_1(x_1), \ldots, p_m(x_m))$ are in R and then $q_j \equiv_i p_j$ holds, for each $1 \leq j \leq m$.

Clearly, there is a $k \geq 0$ such that $\equiv_k$ and $\equiv_{k+j}$ are the same, for every $j \geq 1$. The minimal dttr equivalent to T is defined as $T_{min} = (Q', \Sigma, \Sigma, [q_0]_{\equiv_k}, R')$, where

- *$Q' = \{[q]_{\equiv_k} \mid q \in Q\}$ and*
- *$[q]_{\equiv_k}(\sigma(x_1, \ldots, x_m)) \to \sigma([q_1]_{\equiv_k}(x_1), \ldots, [q_m]_{\equiv_k}(x_m)) \in R'$ if and only if $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, for any $m \geq 0$, $\sigma \in \Sigma_m$, and $q, q_1, \ldots, q_m \in Q$.*

The proof of Proposition 1.4.9 is rather long and needs new concepts to introduce (e.g. dttr congruence, quotient dttr, etc.). However, it is an easy exercise to present it if one follows the proof of Theorem 8 in [GécSte3] step by step. Hence we omit the proof here.

However, we note that, proving Proposition 1.4.9 on the basis of [GécSte3], it should be considered that, in contrast with a dtta, a state of a dttr is not necessarily defined for all input symbols (cf. definitions of $\equiv_{i+1}$ in (ii) of Construction 1.4.10 and ρ_{k+1} in (ii) in [GécSte3] on page 43).

1.4.5 Domain and range tree languages

In this subsection we investigate domain and range tree languages of various types of dt tree transformations from the point of view of recognizability.

Recall that a dttr is also an rl-dt tree transducer, hence $DREC \subseteq \mathrm{dom}(rl\text{-}DT)$ holds. On the other hand, the following statement shows that $\mathrm{dom}(DT) \subseteq DREC$. The original statement can be found as Lemma 5 in [FülVág1] (this result also appears in [Eng2]), although it is slightly modified here for our purposes.

Propositon 1.4.11 *For any dt tree transducer $T = (Q, \Sigma, \Delta, q_0, R)$, a connected dttr $T' = (Q', \Sigma, \Sigma, \{q_0\}, R')$ exists such that $L(T') = dom(\tau_T)$.*

Proof. We define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \ldots$ of subsets of pow(Q) and a sequence $R^{(0)} \subseteq R^{(1)} \subseteq \ldots$ of finite sets of rules of the form $P(\sigma(x_1, \ldots, x_m)) \to \sigma(P_1(x_1), \ldots, P_m(x_m))$, where $m \geq 0$, $\sigma \in \Sigma_m$, $P, P_1, \ldots, P_m \subseteq Q$:

(i) Put $P_0 = \{q_0\}$. Let $Q^{(0)} = \{P_0\} \cup \{\mathrm{rst}(q_0, \sigma, j) \mid m \geq 1, \sigma \in \Sigma_m, 1 \leq j \leq m, q_0$ is defined on σ in $R\}$. Moreover, let $R^{(0)} = \{P_0(\sigma(x_1, \ldots, x_m)) \to \sigma(\mathrm{rst}(q_0, \sigma, 1)(x_1), \ldots, \mathrm{rst}(q_0, \sigma, m)(x_m)) \mid m \geq 0, \sigma \in \Sigma_m, q_0$ is defined on σ in $R\}$.

(ii) Let $k \geq 0$. For any $P \in Q^{(k)}$ and $\sigma \in \Sigma$, we say that P is defined on σ if, for each $q \in P$, q is defined on σ in R and in this case if $\sigma \in \Sigma_m$ with $m \geq 1$, we put $S_{P,\sigma,j} = \bigcup_{q \in P} \mathrm{rst}(q, \sigma, j)$, for every $1 \leq j \leq m$. Specially, $P = \emptyset$ is defined for all $\sigma \in \Sigma$ and if $\sigma \in \Sigma_m$ with $m \geq 1$, then $S_{\emptyset,\sigma,j} = \emptyset$, for each $1 \leq j \leq m$.
Now let $Q^{(k+1)} = Q^{(k)} \cup \{S_{P,\sigma,j} \mid m \geq 1, \sigma \in \Sigma_m, 1 \leq j \leq m, P \in Q^{(k)}$, P is defined on $\sigma\}$. Moreover, let $R^{(k+1)} = R^{(k)} \cup \{P(\sigma(x_1, \ldots, x_m)) \to \sigma(S_{P,\sigma,1}(x_1), \ldots, S_{P,\sigma,m}(x_m)) \mid m \geq 0, \sigma \in \Sigma_m, P \in Q^{(k)}, P$ is defined on $\sigma\}$. Specially, if $\emptyset \in Q^{(k)}$, then $\emptyset(\sigma(x_1, \ldots, x_m)) \to \sigma(\emptyset(x_1), \ldots, \emptyset(x_m)) \in R^{(k+1)}$, for every $m \geq 0$ and $\sigma \in \Sigma_m$.

Clearly, a $k \geq 0$ exists such that $Q^{(k+1)} = Q^{(k)}$, and then $Q^{(k+1)} = Q^{(k+2)} = \ldots$ and $R^{(k+1)} = R^{(k+2)} = \ldots$ hold. Let $Q' = Q^{(k+1)}$ and $R' = R^{(k+1)}$.

It is easy to see that T' is exactly the connected version of the dttr defined in the proof of Lemma 5 in [FülVág1]. □

Observe that if T is linear, then Q' in the proof of Proposition 1.4.11 consists of sets containing at most one element. Moreover, if $\emptyset \in Q'$, then it is a universal state of T'.

We have dom$(rl\text{-}DT)$ = dom$(l\text{-}DT)$ = dom(DT) = $DREC$. Moreover, we recall that $DREC$ =dom(DT^n) =dom$(l\text{-}DT^n)$ for every $n \geq 1$, see [FülVág3].

As for ranges of various types of dt tree transformation classes, we recall the following results.

It is well known and easy to show that range$(nd\text{-}DT)$ is not recognizable and hence range(DT) is also not recognizable. To see this, consider the following example.

Let $T = (Q, \Sigma, \Delta, q_0, R)$ be an nd-dt tree transducer, where

- $Q = \{q_0, q_1\}$,
- $\Sigma = \{\sigma^{(1)}, \#^{(0)}\}$,

- $\Delta = \{\delta^{(2)}, \sigma^{(1)}, \#^{(0)}\}$, and
- $R = \{q_0(\sigma(x_1)) \to \delta(q_1(x_1), q_1(x_1)),\ q_1(\sigma(x_1)) \to \sigma(q_1(x_1)),\ q_1(\#) \to \#\}$.

Then range$(\tau_T) = \{\delta(\sigma^n(\#), \sigma^n(\#)) \mid n \geq 0\}$, which is obviously not recognizable.

On the other hand, by Corollary 6.6 of Chapter IV in [GécSte4], it holds that range$(l\text{-}DT) \subseteq REC$. We note that even the equality can be shown, i.e. range$(l\text{-}DT) = REC$. This result is proved in Section 2.3.

Chapter 2

Properties of sl-dt tree transducers

In this chapter we investigate certain properties of superlinear deterministic top-down tree transducers and tree transformations induced by them.

In Section 2.1 we explore some basic properties, e.g., that *sl-DT* is not closed under the compositions and that $DT = nd\text{-}HOM \circ sl\text{-}DT$ holds. The results of this section appear in Section 3 of [DánFül1].

In Sections 2.2 and 2.3 we investigate domain and range tree languages of superlinear deterministic top-down tree transformations, respectively. We define a new type of top-down tree recognizers, called semi-universal deterministic top-down tree recognizer. We show that the domain tree languages of superlinear deterministic top-down tree transducers are exactly that ones, which are recognized by semi-universal deterministic top-down tree recognizers. On the basis of this result, we develop a decision algorithm, which decides whether an arbitrary deterministic recognizable tree language can be domain of a superlinear deterministic top-down tree transformation. Moreover, we prove that the range tree languages of superlinear deterministic top-down tree transducers are exactly the recognizable tree languages. We note that the results of Sections 2.2 and 2.3 were published in [Dán].

2.1 Basic properties

In this section we describe some basic properties of superlinear deterministic top-down tree transducers and the class *sl-DT*.

First we present two simple observations.

Observation 2.1.1 *Let* $T = (\{q\}, \Sigma, \Delta, q, R)$ *be a hom tree transducer. Then* T *is superlinear if and only if it is linear.*

Proof. By definition, if T is superlinear, then it is also linear. Conversely, let T be linear. Since T has the only state q, for any $\sigma \in \Sigma_m$ with $m \geq 0$ and integer i with $1 \leq i \leq m$, there is at most one rule for σ such that x_i occurs in $\text{rhs}(q, \sigma)$. Hence T is superlinear as well. □

Corollary 2.1.2 *l-HOM = sl-HOM*

The second observation immediately follows from the definitions of nondeleting and superlinear top-down tree transducers.

Observation 2.1.3 *Let $T = (Q, \Sigma, \Delta, q_0, R)$ be a nd-sl-dt tree transducer. For every symbol $\sigma \in \Sigma_n$ with $n \geq 1$, there is at most one state $q \in Q$ such that R contains a (q, σ)-rule.*

In the following theorem, we show that there exist two sl-dt tree transformations such that their composition is not even a dt tree transformation. This implies that the tree transformation class *sl-DT* is not closed under the composition.

Theorem 2.1.4 $sl\text{-}DT^2 - DT \neq \emptyset$

Proof. Consider the ranked alphabets $\Sigma = \{\sigma, \#\}$ and $\Delta = \{\#\}$, where σ and $\#$ have rank 2 and 0, respectively. Define the sl-dt tree transducers T_1 and T_2 in the following way.

Let $T_1 = (Q_1, \Sigma, \Sigma, p, R_1)$ where

- $Q_1 = \{p, q\}$ and
- R consists of the rules $p(\sigma(x_1, x_2)) \to \sigma(q(x_1), q(x_2))$ and $q(\#) \to \#$.

Moreover, let $T_2 = (\{p\}, \Sigma, \Delta, p, \{p(\sigma(x_1, x_2)) \to \#\})$. It is easy to show that $\tau_{T_1} \circ \tau_{T_2} = \{(\sigma(\#, \#), \#)\}$.

Suppose that a dt tree transducer $T = (Q, \Sigma, \Delta, q_0, R)$ exists such that $\tau_T = \tau_{T_1} \circ \tau_{T_2}$. Since $\Delta = \Delta_0$, the only (q_0, σ)-rule in R must be of one of the following three forms:

(1) $q_0(\sigma(x_1, x_2)) \to \#$.

(2) $q_0(\sigma(x_1, x_2)) \to q(x_1)$, for some $q \in Q$.

(3) $q_0(\sigma(x_1, x_2)) \to q(x_2)$, for some $q \in Q$.

We show that each case leads to a contradiction.

Indeed, in case (1), for every $t_1, t_2 \in T_\Sigma$, $(\sigma(t_1, t_2), \#) \in \tau_T$ holds, which is a contradiction.

In case (2), $q(\#) \to \#$ must be in R, because $(\sigma(\#,\#),\#) \in \tau_T$. However, in this case $(\sigma(\#,t),\#) \in \tau_T$, for every $t \in T_\Sigma$, which is a contradiction again.

Similarly, (3) is also impossible. We obtained that a suitable T does not exist. Therefore $\tau_{T_1} \circ \tau_{T_2} \notin DT$. □

We note that, since *sl-DT* is a subclass of *l-DT* and hence of *DT*, the above result also shows that neither *DT* nor *l-DT* is closed under the composition. (However, these results are already known, see, e.g., [FülVág1].)

The property superlinear may seem to be a very strict restriction. Nevertheless, the following lemma and theorem show that the composition of *nd-HOM* and *sl-DT* yields *DT*.

Lemma 2.1.5 *l-DT* $\subseteq$ *nd-HOM* $\circ$ *sl-DT*

Proof. Let $T = (Q, \Sigma, \Delta, q_0, R)$ be an l-dt tree transducer. We define an nd-hom tree transducer $H = (\{p\}, \Sigma, \Sigma', p, R')$ and an sl-dt tree transducer $T' = (Q, \Sigma', \Delta, q_0, R'')$ such that $\tau_T = \tau_H \circ \tau_{T'}$.

To this end, let $p_1, \ldots, p_n$ be a fixed enumeration of the states in Q. Define Σ' to be the smallest ranked alphabet, for which $\Sigma'_{kn} = \{\sigma' \mid \sigma \in \Sigma_k\}$, for every $k \geq 0$. Then let R' be the smallest set of rules satisfying the following condition. For every $k \geq 0$ and $\sigma \in \Sigma_k$, let the rule

$$p(\sigma(x_1, \ldots, x_k)) \to \sigma'(\overbrace{p(x_1), \ldots, p(x_1)}^{n}, \ldots, \overbrace{p(x_k), \ldots, p(x_k)}^{n})$$

be in R'. (Note that x_i occurs n times in $\mathrm{rhs}(p, \sigma)$, for each $1 \leq i \leq k$).

Finally, let R'' be the set of rules constructed as follows. For each $r \in R$, carry on the following procedure.

Assume that r is of the form $q(\sigma(x_1, \ldots, x_k)) \to t$, for some $q \in Q$, $k \geq 0$, and $\sigma \in \Sigma_k$. Suppose that $q = p_i$, for some $1 \leq i \leq n$. Then let the rule $q(\sigma'(x_1, \ldots, x_{kn})) \to t'$ be in R'', where the tree t' is obtained from t in the way that, for every $1 \leq j \leq k$, we substitute x_j by $x_{(j-1)n+i}$ in t. (More formally, we put $t' = t[x_i, x_{n+i}, \ldots, x_{(k-1)\cdot n+i}]$.) Let R'' consist of only these rules.

We show that T' is superlinear. Since T is linear and, by the above construction, different variables in the right-hand side of a rule in R are substituted by different ones, T' is certainly linear.

Moreover, consider a (q, σ')-rule and a (q', σ')-rule from R'', where $\sigma' \in \Sigma'_{kn}$, for some $k \geq 0$, and $q, q' \in Q$ are different states. Suppose that the state q is the ith one and the state q' is the jth one in the sequence $p_1, \ldots, p_n$. Obviously, $i \neq j$ and $1 \leq i, j \leq n$. Then, by the construction of R'', $\mathrm{var}(\mathrm{rhs}(q, \sigma')) \subseteq \{x_i, x_{n+i}, \ldots, x_{(k-1)n+i}\}$ and $\mathrm{var}(\mathrm{rhs}(q', \sigma')) \subseteq \{x_j, x_{n+j}, \ldots, x_{(k-1)n+j}\}$. By $i \neq j$,

$$\{x_i, x_{n+i}, \ldots, x_{(k-1)n+i}\} \cap \{x_j, x_{n+j}, \ldots, x_{(k-1)n+j}\} = \emptyset$$

holds, that is $\mathrm{var}(\mathrm{rhs}(q, \sigma')) \cap \mathrm{var}(\mathrm{rhs}(q', \sigma')) = \emptyset$. Hence T' is superlinear.

To prove that $\tau_T = \tau_H \circ \tau_{T'}$ holds, it is enough to show the following equivalence. For every $s \in T_\Sigma, t \in T_\Delta$ and $q \in Q$, $q(s) \Rightarrow_T^* t$ if and only if an $s' \in T_{\Sigma'}$ exists such that $p(s) \Rightarrow_H^* s'$ and $q(s') \Rightarrow_{T'}^* t$. However, this is straightforward to show by induction on the height of s, hence we do not present the formal proof. $\square$

We can easily derive the following two very important results from Lemma 2.1.5. The first one shows that any dt tree transformation can be substituted by the consecutive application of an nd-dt and an sl-dt tree transducer, cf. Proposition 1.4.4.

Theorem 2.1.6 *DT = nd-HOM ∘ sl-DT*

Proof.

$$\begin{array}{llll} DT & = & \textit{nd-HOM} \circ \textit{l-DT} & \text{(by Proposition 1.4.4)} \\ & \subseteq & \textit{nd-HOM} \circ \textit{nd-HOM} \circ \textit{sl-DT} & \text{(by Lemma 2.1.5)} \\ & = & \textit{nd-HOM} \circ \textit{sl-DT} & \text{(by (3) of Corollary 1.4.2)} \\ & \subseteq & DT & \text{(by Proposition 1.4.4)} \end{array} \quad \square$$

By Proposition 1.4.3, any tree transformation, given by the composition of an arbitrary sequence of deterministic top-down tree transformations, can be substituted by the consecutive application of two appropriate dt tree transducers.

The following result show that, informally speaking, the subsequent application of an nd-hom and two sl-dt tree transducers has the same transformation power.

Corollary 2.1.7 $DT^2 = \textit{nd-HOM} \circ \textit{sl-DT}^2$

Proof.

$$\begin{array}{llll} DT^2 & = & DT \circ \textit{nd-HOM} \circ \textit{sl-DT} & \text{(by Theorem 2.1.6)} \\ & = & DT \circ \textit{sl-DT} & \text{(by (5) of Corollary 1.4.2)} \\ & = & \textit{nd-HOM} \circ \textit{sl-DT}^2 & \text{(by Theorem 2.1.6)} \end{array} \quad \square$$

By definition, $\mathrm{var}(\mathrm{rhs}(q,\sigma)) \cap \mathrm{var}(\mathrm{rhs}(q',\sigma)) = \emptyset$, for any (q,σ)-, and (q',σ)-rules of an sl-dt tree transducer $T = (Q, \Sigma, \Delta, q_0, R)$, if $q \neq q'$.

In the following technical lemma we show that this property is hereditary for trees r and r' with variables, which are obtained by derivation from the same tree s starting with some different states q and q', respectively.

Lemma 2.1.8 *Let $T = (Q, \Sigma, \Delta, q_0, R)$ be an sl-dt tree transducer. Let $s \in \hat{T}_{\Sigma,k}$, for some $k \geq 1$, and let $q, q' \in Q$ be different states such that the derivations $q(s) \Rightarrow_T^* r$ and $q'(s) \Rightarrow_T^* r'$ hold for some trees $r, r' \in T_\Delta(Q(X_k))$. Then $var(r) \cap var(r') = \emptyset$ if and only if $s \neq x_1$.*

Proof. In other words, our lemma states that, for every nontrivial tree $s \in \hat{T}_{\Sigma,k}$ with $k \geq 1$ and for every $1 \leq i \leq k$, there is at most one state $q \in Q$ such that x_i occurs in r, where r is defined by $q(s) \Rightarrow_T^* r$.

Clearly, if $s = x_1$ then $r = q(x_1)$ and $r' = q'(x_1)$, hence $\mathrm{var}(r) \cap \mathrm{var}(r') = \{x_1\} \neq \emptyset$.

Now let $s \neq x_1$, that is $s = \sigma(s_1, \ldots, s_m)$, for some $m \geq 0$, $\sigma \in \Sigma_m$, and $s_1, \ldots, s_m \in T_{\Sigma,k}$. Then the derivations $q(s) \Rightarrow_T^* r$ and $q'(s) \Rightarrow_T^* r'$ can be detailed as

$$q(s) = q(\sigma(s_1, \ldots, s_m)) \underset{T}{\Rightarrow} t[q_1(s_{i_1}), \ldots, q_n(s_{i_n})] \overset{*}{\underset{T}{\Rightarrow}} t[t_1, \ldots, t_n] = r$$

and

$$q'(s) = q'(\sigma(s_1, \ldots, s_m)) \underset{T}{\Rightarrow} t'[p_1(s_{j_1}), \ldots, p_l(s_{j_l})] \overset{*}{\underset{T}{\Rightarrow}} t'[t'_1, \ldots, t'_l] = r',$$

respectively, where the rules

$$q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$$

and

$$q'(\sigma(x_1, \ldots, x_m)) \to t'[p_1(x_{j_1}), \ldots, p_l(x_{j_l})]$$

are in R.

Since $s \in \hat{T}_{\Sigma,k}$, it holds that $\mathrm{var}(s_i) \cap \mathrm{var}(s_j) = \emptyset$, whenever $1 \leq i \neq j \leq m$. On the other hand, by the superlinear property of T, we have $\{i_1, \ldots, i_n\} \cap \{j_1, \ldots, j_l\} = \emptyset$.

Consequently, we have $\mathrm{var}(s_{i_u}) \cap \mathrm{var}(s_{j_v}) = \emptyset$ for every integer $1 \leq u \leq n$ and $1 \leq v \leq l$. Moreover, any variable occurring in t_u (resp. t'_v) also occurs in s_{i_u} (resp. s'_{j_v}). Therefore, we also have $\mathrm{var}(t_u) \cap \mathrm{var}(t'_v) = \emptyset$, which proves $\mathrm{var}(r) \cap \mathrm{var}(r') = \emptyset$. □

The following decomposition lemma plays a very important role in the proof of the main result of Section 3.1.

Lemma 2.1.9 *DT = op-ni-DT ∘ nr-l-nd-HOM*

Proof. The inclusion *op-ni-DT ∘ nr-l-nd-HOM ⊆ DT* should be clear by (5) of Corollary 1.4.2.

To prove the converse let $T = (Q, \Sigma, \Delta, q_0, R)$ be an arbitrary dt tree transducer. We show that an op-ni-dt tree transducer T' and an nr-l-nd-hom tree transducer T'' exist so that $\tau_T = \tau_{T'} \circ \tau_{T''}$.

We construct the op-ni-dt tree transducer $T' = (Q, \Sigma, \Sigma', q_0, R')$ and the nr-l-nd-hom tree transducer $T'' = (\{p\}, \Sigma', \Delta, p, R'')$ in the following way. Define the ranked alphabet Σ' such that

$$\Sigma'_n = \{\sigma^q \mid q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})] \in R \text{ and } t \neq x_1\},$$

for every $n \geq 0$.

Then define the sets R' and R'' of rules as the smallest sets satisfying the following conditions. Let $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ be a rule in R, where $t \neq x_1$. Let $j_1, \ldots, j_n$ be a permutation of the numbers $1, \ldots, n$, for which $i_{j_1} \leq \ldots \leq i_{j_n}$. (Clearly, at least one such permutation exists. If there are more possibilities, then fix one.) Let $k_1, \ldots, k_n$ be the inverse of the permutation $j_1, \ldots, j_n$. Hence $j_{k_l} = l$ holds for each l such that $1 \leq l \leq n$. Then let

$$q(\sigma(x_1, \ldots, x_m)) \to \sigma^q(q_{j_1}(x_{i_{j_1}}), \ldots, q_{j_n}(x_{i_{j_n}}))$$

be in R' and let

$$p(\sigma^q(x_1, \ldots, x_n)) \to t[p(x_{k_1}), \ldots, p(x_{k_n})]$$

be in R''.

Additionally, for every reducing rule $q(\sigma(x_1, \ldots, x_m)) \to q'(x_i) \in R$, let that rule be also in R'.

Clearly, in this case T' is op-ni-dt and T'' is nr-l-nd-hom. Moreover, it is easy to verify that $\tau_T = \tau_{T'} \circ \tau_{T''}$. Hence $DT \subseteq$ *op-ni-DT* $\circ$ *nr-l-dn-HOM*. $\square$

Moreover, we show that the total version of the previous theorem also holds.

Corollary 2.1.10 *t-DT* $=$ *t-op-ni-DT* $\circ$ *nr-l-nd-HOM*

Proof. It is sufficient to observe that if T in the proof of Lemma 2.1.9 is total, then T' will be total as well. $\square$

We know (see Proposition 1.4.1) that the syntactic composition of dt tree transducers preserves any of the properties t, l, nd, and hom. We have studied this problem for the sl property and have obtained the following result.

Lemma 2.1.11 *The syntactic composition $T'' = T \circ T'$ of two sl-dt tree transducers $T = (Q, \Sigma, \Delta, q_0, R)$ and $T' = (Q', \Delta, \Omega, q_0', R')$ is an sl-dt tree transducer if and only if T is nonreducing or Q' is a singleton set.*

Proof. We recall that $T'' = (Q' \times Q, \Sigma, \Omega, (q_0', q_0), R'')$, where R'' is defined in the way described in Subsection 2.5. Moreover, by (3) of Proposition 1.4.1, T'' is linear.

First suppose that T does not have the nonreducing property and Q' is not a singleton set. This means that there is a reducing rule $q(\sigma(x_1, \ldots, x_m)) \to q_1(x_i)$ in R and there are two different states p and p' in Q'. In this case, by the definition of R'', both the rules

$$(p, q)(\sigma(x_1, \ldots, x_m)) \to (p, q_1)(x_i)$$

and

$$(p', q)(\sigma(x_1, \ldots, x_m)) \to (p', q_1)(x_i)$$

are in R''. Then x_i appears both in $\mathrm{rhs}((p,q),\sigma)$ and $\mathrm{rhs}((p',q),\sigma)$, consequently T'' is not an sl-dt transducer.

Next, suppose that $Q' = \{p\}$ is a singleton set. Then, for every rule

$$q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})] \in R,$$

at most one rule of the form

$$(p, q)(\sigma(x_1, \ldots, x_m)) \to t'[(p, q_{j_1})(x_{i_{j_1}}), \ldots, (p, q_{j_k})(x_{i_{j_k}})]$$

can be obtained for R', where the right-hand side is determined by the conditions

$$p(t) \underset{T'}{\overset{*}{\Rightarrow}} t'[p(x_{j_1}), \ldots, p(x_{j_k})]$$

and $t' \in \hat{T}_{\Delta,k}$. Moreover, by inspection, $\mathrm{var}(\mathrm{rhs}((p,q),\sigma)) \subseteq \mathrm{var}(\mathrm{rhs}(q,\sigma))$. Consequently, if q and q' are different states in Q, then

$$\mathrm{var}(\mathrm{rhs}((p,q),\sigma)) \cap \mathrm{var}(\mathrm{rhs}((p,q'),\sigma)) \subseteq \mathrm{var}(\mathrm{rhs}(q,\sigma)) \cap \mathrm{var}(\mathrm{rhs}(q',\sigma)) = \emptyset,$$

showing that T'' is superlinear.

Finally, assume that T is an nr-sl-dt tree transducer. We must show that

$$\mathrm{var}(\mathrm{rhs}((p,q),\sigma)) \cap \mathrm{var}(\mathrm{rhs}((p',q'),\sigma)) = \emptyset,$$

for all states $p, p' \in Q'$ and $q, q' \in Q$ such that $p \neq p'$ or $q \neq q'$.

In the case when $q \neq q'$ our statement follows from

$$\mathrm{var}(\mathrm{rhs}(q,\sigma)) \cap \mathrm{var}(\mathrm{rhs}(q',\sigma)) = \emptyset$$

by an argumentation similar to the previous one.

Thus, the only case we have to deal with is $q = q'$ and $p \neq p'$. Therefore, take a rule

$$q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})] \in R, \qquad (*)$$

where $t \neq x_1$ holds by the nonreducing property. Moreover, suppose that

$$p(t) \underset{T'}{\overset{*}{\Rightarrow}} t'[p_1(x_{j_1}), \ldots, p_k(x_{j_k})]$$

and

$$p'(t) \underset{T'}{\overset{*}{\Rightarrow}} t''[p'_1(x_{j'_1}), \ldots, p'_l(x_{j'_l})].$$

In this case both the rules

$$(p, q)(\sigma(x_1, \ldots, x_m)) \to t'[(p_1, q_{j_1})(x_{i_{j_1}}), \ldots, (p_k, q_{j_k})(x_{i_{j_k}})]$$

and

$$(p', q)(\sigma(x_1, \ldots, x_m)) \to t''[(p'_1, q_{j'_1})(x_{i_{j'_1}}), \ldots, (p'_l, q_{j'_l})(x_{i_{j'_l}})]$$

are in R''. Since T' is superlinear, we have $\{j_1, \ldots, j_k\} \cap \{j'_1, \ldots, j'_l\} = \emptyset$, by Lemma 2.1.8.

On the other hand, since T is linear as well, the integers $i_1, \ldots, i_n$ in $(*)$ are pairwise different. Hence we have also $\{i_{j_1}, \ldots, i_{j_k}\} \cap \{i_{j'_1}, \ldots, i_{j'_l}\} = \emptyset$, which means that $\mathrm{var}(\mathrm{rhs}((p,q),\sigma)) \cap \mathrm{var}(\mathrm{rhs}((p',q),\sigma)) = \emptyset$. Thus T'' is superlinear in this case, too. □

Many equations regarding *sl-DT* can be derived from the results of Lemma 2.1.9, Corollary 2.1.10, and Lemma 2.1.11. We list that ones, which will be used in what follows.

Corollary 2.1.12

(1) t-sl-DT	=	*t-sl-DT* ∘ *l-HOM*
(2) sl-DT	=	*sl-DT* ∘ *l-nd-HOM*
(3) sl-DT	=	*t-nr-sl-DT* ∘ *sl-DT*
(4) sl-DT	=	*op-ni-sl-DT* ∘ *nr-l-nd-HOM*
(5) t-sl-DT	=	*t-op-ni-sl-DT* ∘ *nr-l-nd-HOM*
(6) sl-DT	=	*nr-l-nd-HOM* ∘ *sl-DT*
(7) op-ni-sl-DT	=	*t-nr-op-ni-sl-DT* ∘ *op-ni-sl-DT*

Proof.

(1) Since $I \subseteq$ *l-HOM*, the inclusion *t-sl-DT* ⊆ *t-sl-DT* ∘ *l-HOM* should be obvious. To show its converse, consider a t-sl-dt tree transducer T', an l-hom tree transducer T'' and the tree transducer $T = T' \circ T''$. Since T' and T'' are total, T is total as well, by (3) of Proposition 1.4.1. Moreover, $\tau_T = \tau_{T'} \circ \tau_{T''}$ holds, by (1) of the same proposition. Both T' and T'' are superlinear (see Observation 2.1.1) and, being a hom tree transducer, T'' has a singleton state set. Hence T is also superlinear by Lemma 2.1.11. With this we proved that the inclusion *t-sl-DT* ∘ *l-HOM* ⊆ *t-sl-DT* also holds.

(2) The inclusion *sl-DT* ⊆ *sl-DT* ∘ *l-nd-HOM* obviously holds. Conversely, let T' be an sl-dt tree transducer, let T'' be an nd-l-hom tree transducer and put $T = T' \circ T''$. Then, by Lemma 2.1.11, T is superlinear. Moreover, by (1) of Proposition 1.4.1, $\tau_T = \tau_{T'} \circ \tau_{T''}$ holds. Hence the reversed inclusion.

(3) The inclusion *sl-DT* ⊆ *t-nr-sl-DT* ∘ *sl-DT* should be clear. Let T' be a t-nr-sl-dt tree transducer, let T'' be an sl-dt tree transducer and let $T = T' \circ T''$. Then T is superlinear by Lemma 2.1.11. Moreover, $\tau_T = \tau_{T'} \circ \tau_{T''}$ holds, by (1) of Proposition 1.4.1. Hence the reversed inclusion.

(4) The inclusion *op-ni-sl-DT* ∘ *nr-l-nd-HOM* ⊆ *sl-DT* holds by (5) of Corollary 1.4.2 and by Lemma 2.1.11. Conversely, let T be an sl-dt tree transducer. Let the op-ni-dt tree transducer T' and the nr-l-nd-hom tree transducer T'' be

constructed from T in the same way as in the proof of Lemma 2.1.9. Observe, that T' will be superlinear, by that construction. Hence $sl\text{-}DT \subseteq op\text{-}ni\text{-}sl\text{-}DT \circ nr\text{-}l\text{-}nd\text{-}HOM$ holds.

(5) This should be clear by (4) and by Corollary 2.1.10.

(6) It holds that $I \subseteq nr\text{-}l\text{-}nd\text{-}HOM$, hence $sl\text{-}DT \subseteq nr\text{-}l\text{-}nd\text{-}HOM \circ sl\text{-}DT$. Moreover, since $nr\text{-}l\text{-}nd\text{-}HOM \subseteq sl\text{-}DT$ (see Corollary 2.1.2), the inclusion $nr\text{-}l\text{-}nd\text{-}HOM \circ sl\text{-}DT \subseteq sl\text{-}DT$ is obvious by Lemma 2.1.11 and Proposition 1.4.1.

(7) Letting $C = t\text{-}nr\text{-}op\text{-}ni\text{-}sl\text{-}DT \circ op\text{-}ni\text{-}sl\text{-}DT$, we have $op\text{-}ni\text{-}sl\text{-}DT \subseteq C$ by $I \subseteq t\text{-}nr\text{-}op\text{-}ni\text{-}sl\text{-}DT$. Moreover, $C \subseteq sl\text{-}DT$ holds by Lemma 2.1.11. On the other hand, by the proof of that lemma, it is easy to see that $C \subseteq op\text{-}ni\text{-}sl\text{-}DT$ holds as well. □

Theorem 2.1.4 shows that there are two sl-dt tree transformations such that their composition cannot be induced by a dt and hence by an l-dt tree transducer. This suggests that the consecutive application of a sequence of sl-dt tree transducers has big transformation power, that is, for instance, any l-dt or dt tree transformation can be represented as composition of sl-dt tree transformations.

However, the following theorem shows that this is not the case. Namely, we show that generally even the total l-dt tree transformations cannot be induced by sequences of sl-dt tree transformations.

Theorem 2.1.13 $t\text{-}l\text{-}DT - sl\text{-}DT^+ \neq \emptyset$

Proof. Let the t-l-dt tree transducer $T = (Q, \Sigma, \Delta, q, R)$ be defined as follows:

- $Q = \{q, q'\}$.
- $\Sigma = \{\sigma^{(1)}, \#^{(0)}\}$.
- $\Delta = \Sigma \cup \{\$^{(0)}\}$.
- $R = \{q(\sigma(x_1)) \to \sigma(q'(x_1)),\ q'(\sigma(x_1)) \to \sigma(q(x_1)),\ q(\#) \to \#,\ q'(\#) \to \$\}$.

It should be clear that

$$\tau_T = \{(\sigma^m(\#), \sigma^m(\#)) \mid m \geq 0 \text{ is even}\} \cup \{(\sigma^m(\#), \sigma^m(\$)) \mid m \geq 0 \text{ is odd}\}.$$

We prove by contradiction that $\tau_T \notin sl\text{-}DT^n$ for any $n \geq 1$.

Therefore, suppose that an integer $n \geq 1$ and sl-dt tree transducers $T_1, \ldots, T_n$ exist such that $\tau_T = \tau_{T_1} \circ \cdots \circ \tau_{T_n}$ holds. Without loss of generality, we can assume that n is chosen to be minimal. We put $\tau_n = \tau_{T_1} \circ \cdots \circ \tau_{T_n}$.

Let $T_1 = (Q_1, \Sigma, \Sigma', q_1, R_1)$. Observe that, since $(\#, \#) \in \tau_T$, there must be a $(q_1, \#)$-rule in R_1 of the form

$$q_1(\#) \to s,$$

where $s \in T_{\Sigma'}$.

Similarly, $\tau_T(\sigma(\#)) = \sigma(\$)$ implies that there must be a (q_1, σ)-rule in R_1. Moreover, $\mathrm{rhs}(q_1, \sigma)$ must contain the variable x_1. Otherwise, $\mathrm{rhs}(q_1, \sigma) \in T_{\Sigma'}$ implies $\tau_{T_1}(\sigma(\#)) = \tau_{T_1}(\sigma^2(\#))$ and hence $\tau_n(\sigma(\#)) = \tau_n(\sigma^2(\#))$, which contradicts $\tau_T(\sigma^2(\#)) = \sigma^2(\#)$.

Since T_1 is linear, the (q_1, σ)-rule is of the form

$$q_1(\sigma(x_1)) \to r[p(x_1)],$$

for some $r \in \hat{T}_{\Sigma',1}$ and $p \in Q_1$.

Since $\tau_T(\sigma^2(\#)) = \sigma^2(\#)$, there must be a (p, σ)-rule in R_1, too. Assume that $p \neq q_1$. Then $\mathrm{rhs}(p, \sigma)$ cannot contain x_1, because $\mathrm{rhs}(q_1, \sigma)$ already does so and T_1 is superlinear. However, this implies $\tau_{T_1}(\sigma^2(\#)) = \tau_{T_1}(\sigma^3(\#))$, which contradicts $\tau_T(\sigma^3(\#)) = \sigma^3(\$)$. Therefore, $p = q_1$ holds and thus the rule

$$q_1(\sigma(x_1)) \to r[q_1(x_1)]$$

is in R_1.

We can observe that only the two q_1-rules

$$q_1(\#) \to s$$

and

$$q_1(\sigma(x_1)) \to r[q_1(x_1)]$$

can be useful in any derivation using T_1, where $s \in T_{\Sigma'}$ and $r \in \hat{T}_{\Sigma',1}$. Hence T_1 is total.

Next we show that the tree $r \in \hat{T}_{\Sigma,1}$ appearing in the above (q_1, σ)-rule cannot be x_1, that is the rule cannot be a reducing one.

For if $r = x_1$, i.e. $q_1(\sigma(x_1)) \to q_1(x_1)$ is in R_1, then $q_1(\sigma^m(\#)) \Rightarrow^*_{T_1} s$ holds, for every $m \geq 0$, implying $|\mathrm{range}(\tau_n)| = 1$, which is obviously not true. Hence T_1 is nonreducing and thus it is a t-nr-sl-dt tree transducer.

By (3) of Corollary 2.1.12 and by (1) of Proposition 1.4.1, the tree transducer $T_2' = T_1 \circ T_2$ is superlinear and $\tau_{T_2'} = \tau_{T_1} \circ \tau_{T_2}$ holds. Hence we have $\tau_T = \tau_{T_2'} \circ \tau_{T_3} \circ \ldots \circ \tau_{T_n}$, which contradicts our assumption that n is the smallest integer such that τ_T is a composition of n sl-dt tree transformations. □

Considering Theorem 2.1.13, we can easily show the following inclusions, which proves to be very useful in later chapters.

Corollary 2.1.14

(1)	*sl-DT*^{n}	$\subset$	*l-DT*^{n}*, for every* $n \geq 1$.
(2)	*sl-DT*$^+$	$\subset$	*l-DT*^{2}.
(3)	*t-sl-DT*$^+$	$\subset$	*t-l-DT*.

Proof. We recall $l\text{-}DT \subset l\text{-}DT^2$ (see [Rou]) and $l\text{-}DT^n = l\text{-}DT^2$, for $n \geq 2$, see Proposition 1.4.3.

(1) Clearly, $sl\text{-}DT^n \subseteq l\text{-}DT^n$. On the other hand, $sl\text{-}DT^n = l\text{-}DT^n$, for some $n \geq 1$, would imply $t\text{-}l\text{-}DT \subset sl\text{-}DT^n$, which contradicts Theorem 2.1.13.

(2) We have $sl\text{-}DT^+ \subseteq l\text{-}DT^+ = l\text{-}DT^2$. Clearly, $sl\text{-}DT^+ = l\text{-}DT^2$ implies $t\text{-}l\text{-}DT \subset sl\text{-}DT^+$ contradicting again Theorem 2.1.13.

(3) By Proposition 1.4.1, it is easy to see that $t\text{-}l\text{-}DT^+ = t\text{-}l\text{-}DT$, hence $t\text{-}st\text{-}DT^+ \subseteq t\text{-}l\text{-}DT$ holds. Moreover, the proper inclusion immediately follows from Theorem 2.1.13. □

2.2 Domain tree languages

In this section we give a characterization of the class dom($sl\text{-}DT$). Moreover, we show that, for any $L \in DREC$, it is decidable whether $L \in$ dom($sl\text{-}DT$) holds and we present also a decision procedure.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. We say that T is a *semi-universal deterministic top-down tree recognizer (su-dttr)*, if the following condition holds. For any $m \geq 1$, $\sigma \in \Sigma_m$, and two different states $q, p \in Q$, if $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma(p_1(x_1), \ldots, p_m(x_m))$ are in R, then, for each $1 \leq i \leq m$, at least one of q_i and p_i is universal. We denote by $su\text{-}DREC$ the class of tree languages recognized by su-dttr's.

First we show that dom($sl\text{-}DT$) and $su\text{-}DREC$ are equal classes.

Lemma 2.2.1 *For any sl-dt tree transducer* $T = (Q, \Sigma, \Delta, q_0, R)$, *dom*$(\tau_T)$ *is recognized by an su-dttr.*

Proof. Let the dttr $T' = (Q', \Sigma, \Sigma, \{q_0\}, R')$ be constructed from T as defined in the proof of Proposition 1.4.11, then $L(T') = \text{dom}(\tau_T)$. We show that T' is an su-dttr.

Since T is linear, each set in Q' contains at most one element. Observe that, for any $m \geq 1$, $\sigma \in \Sigma_m$, and $q \in Q$, if $\{q\}(\sigma(x_1, \ldots, x_m)) \to \sigma(P_1(x_1), \ldots, P_m(x_m))$ is in R', then $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})] \in R$, for some $0 \leq n \leq m$ and $t \in \hat{T}_{\Delta,n}$. Moreover, for each $1 \leq j \leq m$, if $j = i_k$, for some $1 \leq k \leq n$, then $P_j = \{q_k\}$, and $P_j = \emptyset$ otherwise. Note that, since T is linear, the i_ks are different.

Now suppose that, for some $m \geq 1$, $\sigma \in \Sigma_m$ and two different states $q, p \in Q$, $\{q\}(\sigma(x_1, \ldots, x_m)) \to \sigma(P_{q,1}(x_1), \ldots, P_{q,m}(x_m))$ and $\{p\}(\sigma(x_1, \ldots, x_m)) \to \sigma(P_{p,1}(x_1), \ldots, P_{p,m}(x_m))$ are in R'. Then, by the above observations, there exist rules $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ and $p(\sigma(x_1, \ldots, x_m)) \to s[p_1(x_{i'_1}), \ldots, p_{n'}(x_{i'_{n'}})]$ in R, for some $0 \leq n, n' \leq m$, $t \in \hat{T}_{\Delta,n}$, and $s \in \hat{T}_{\Delta,n'}$, where, for each $1 \leq j \leq m$, if $j = i_k$, for some $1 \leq k \leq n$, then $P_{q,j} = \{q_k\}$, else

$P_{q,j} = \emptyset$. Moreover, if $j = i'_k$, for some $1 \le k \le n'$, then $P_{p,j} = \{p_k\}$, and $P_{p,j} = \emptyset$ otherwise.

Since T is superlinear, $\{i_1, \ldots, i_n\} \cap \{i'_1, \ldots, i'_{n'}\} = \emptyset$, hence we have that, for each $1 \le j \le m$, at least one of $P_{q,j}$ and $P_{p,j}$ is $\emptyset$. We saw that if $\emptyset \in Q'$, then it is necessarily a universal state, hence T' is an su-dttr. □

Lemma 2.2.2 *For any $L \in$ su-DREC, an sl-dt tree transducer T' exists such that $dom(\tau_{T'}) = L$.*

Proof. Suppose that $L \in$ *su-DREC*, then it is recognized by an su-dttr $T = (Q, \Sigma, \Sigma, q_0, R)$. For every $m \ge 0$, $\sigma \in \Sigma_m$, $q \in Q$, and $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, consider the set $\{i_1, \ldots, i_n\} \subseteq \{1, \ldots, m\}$ of indices, where $i_1 < \ldots < i_n$ and, for any $1 \le j \le m$, $j \in \{i_1, \ldots, i_n\}$ holds if and only if q_j is not a universal state. Then let σ_q be a new symbol having the rank n and define the rule $r_{q,\sigma} : q(\sigma(x_1, \ldots, x_m)) \to \sigma_q(q_{i_1}(x_{i_1}), \ldots, q_{i_n}(x_{i_n}))$.

Let $T' = (Q, \Sigma, \Delta, q_0, R')$ be a dt transducer, where

- $\Delta = \{\sigma_q \mid q \text{ is defined on } \sigma \text{ in } R\}$ and
- $R' = \{r_{q,\sigma} \mid q \text{ is defined on } \sigma \text{ in } R\}$.

We show that T' is superlinear. Obviously, it is linear. Let $q, p \in Q$ be two different states. Suppose that $q(\sigma(x_1, \ldots, x_m)) \to \sigma_q(q_1(x_{i_1}), \ldots, q_k(x_{i_k}))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma_p(p_1(x_{j_1}), \ldots, p_l(x_{j_l}))$ are in R, for some $m \ge 0$, $0 \le k, l \le m$ and $\sigma \in \Sigma_m$. Then, by the construction of T', $\{q_{i_1}, \ldots, q_{i_k}\}$ and $\{q_{j_1}, \ldots, q_{j_l}\}$ are the sets of non-universal states of rhs(q, σ) and rhs(p, σ) in T, respectively. Since T is an su-dttr, $\{i_1, \ldots, i_k\} \cap \{j_1, \ldots, j_l\} = \emptyset$ holds. Therefore T' is superlinear.

Finally, we show that, for any tree $t \in T_\Sigma$ and state $q \in Q$, $q(t) \Rightarrow_T^* t$ holds if and only if $q(t) \Rightarrow_{T'}^* t'$, for some $t' \in T_\Delta$. This implies dom$(\tau_T) = L$ immediately. We prove the statement by induction on height(t).

Basis. Suppose that height$(t) = 0$, then $t = \delta$, for some $\delta \in \Sigma_0$. By the definition of T', $q(\delta) \to \delta_q \in R'$ if and only if $q(\delta) \to \delta \in R$, hence the statement holds by $t' = \delta_q$.

Induction step. Suppose that height$(t) = n + 1$ with $n \ge 0$, then $t = \sigma(t_1, \ldots, t_m)$, for some $m \ge 1$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in T_\Sigma$, where height$(t_i) \le n$, for each $1 \le i \le m$.

Recall that $q(\sigma(x_1, \ldots, x_m)) \to \sigma_q(q_{i_1}(x_{i_1}), \ldots, q_{i_n}(x_{i_n})) \in R'$ if and only if $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, where $q_{i_1}, \ldots, q_{i_n}$ are exactly the non-universal states of rhs(q, σ) in T. Furthermore, by the induction hypothesis, for each $j \in \{i_1, \ldots, i_n\}$, $q_j(t_j) \Rightarrow_T^* t_j$ holds if and only if $q_j(t_j) \Rightarrow_{T'}^* t'_j$, for some $t'_j \in T_\Delta$. Hence $q(t) \Rightarrow_T \sigma(q_1(t_1), \ldots, q_m(t_m)) \Rightarrow_T^* t$ if and only if $q(t) \Rightarrow_{T'} \sigma_q(q_{i_1}(t_{i_1}), \ldots, q_{i_n}(t_{i_n})) \Rightarrow_{T'}^* t'$, where $t' = \sigma_q(t'_{i_1}, \ldots, t'_{i_n})$. □

Summarizing the results of the above two lemmas, we have that the domain tree languages of sl-dt tree transformations are exactly those tree languages, which are recognized by su-dttr's.

Theorem 2.2.3 $dom(sl\text{-}DT) = su\text{-}DREC$

In the rest of the section we show that, for any $L \in DREC$ given by a dttr recognizing L, it is decidable whether $L \in \mathrm{dom}(sl\text{-}DT)$ holds. Moreover, we present a decision procedure.

Recall that, for a dttr T, T_{nor} and T_{con} denote the normalized and connected equivalents of T, according to Propositions 1.4.7 and 1.4.8, respectively. Moreover, if T is normalized and connected, then T_{min} is the minimal equivalent of T, according to Proposition 1.4.9.

Lemma 2.2.4 *Let* $T = (Q, \Sigma, \Sigma, q_0, R)$ *be an su-dttr, then* T_{nor} *and* T_{con} *are su-dttr's, too. Moreover, if* T *is normalized and connected, then* T_{min} *is also an su-dttr.*

Proof. By Proposition 1.4.7, $T_{nor} = (Q', \Sigma, \Sigma, q_0, R')$, where $Q' \subseteq Q$ and $R' \subseteq R$ hold. Hence it should be clear that if T is an su-dttr, then T_{nor} is also an su-dttr.

Similarly, by Proposition 1.4.8, $T_{con} = (Q'', \Sigma, \Sigma, q_0, R'')$, where $Q'' \subseteq Q$ and $R'' \subseteq R$. Therefore if T is an su-dttr, then T_{con} is necessarily an su-dttr, too.

Now suppose that T is a normalized and connected su-dttr. Denote by $\equiv$ the equivalence relation, by which T_{min} is constructed from T (see Construction 1.4.10), that is $T_{min} = (Q''', \Sigma, \Sigma, [q_0]_\equiv, R''')$, where $R''' = \{[q]_\equiv(\sigma(x_1, \ldots, x_m)) \to \sigma([q_1]_\equiv(x_1), \ldots, [q_m]_\equiv(x_m)) \mid q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R\}$ and $Q''' = \{[q]_\equiv \mid q \in Q\}$.

It can be easily shown that if T has universal states, then they constitute exactly one class of Q with respect to $\equiv$. Moreover, if this class exists, then it is the only universal state in T_{min}. By the construction of $\equiv$, the proofs of these statements are straightforward.

Suppose that the states $q, p \in Q$ are in different classes with respect to $\equiv$, that is $[q]_\equiv \neq [p]_\equiv$. If, for some $\sigma \in \Sigma_m$ with $m \geq 1$, both $[q]_\equiv$ and $[p]_\equiv$ are defined on σ in R''', then the $([q]_\equiv, \sigma)$-rule and the $([p]_\equiv, \sigma)$-rule of R''' can be written of the form $[q]_\equiv(\sigma(x_1, \ldots, x_m)) \to \sigma([q_1]_\equiv(x_1), \ldots, [q_m]_\equiv(x_m))$ and $[p]_\equiv(\sigma(x_1, \ldots, x_m)) \to \sigma([p_1]_\equiv(x_1), \ldots, [p_m]_\equiv(x_m))$, respectively, where $q_1, \ldots, q_m, p_1, \ldots, p_m \in Q$ and the rules $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma(p_1(x_1), \ldots, p_m(x_m))$ are in R. Since T is an su-dttr, then, for any $1 \leq i \leq m$, at least one of q_i and p_i is universal in T. Thus, by the observations of the previous paragraph, at least one of $[q_i]_\equiv$ and $[p_i]_\equiv$ is universal in T_{min}. Therefore T_{min} is an su-dttr, too. $\square$

We recall from Proposition 1.4.9 that, for any $L \in DREC$, the minimal dttr recognizing L is unique up to isomorphism. Denote this dttr by T_L. The following theorem establishes our decidability result.

Lemma 2.2.5 *For any tree language $L \in DREC$, $L \in su\text{-}DREC$ if and only if T_L is an su-dttr.*

Proof. If T_L is an su-dttr, then $L \in su\text{-}DREC$ by definition. Conversely, suppose $L \in su\text{-}DREC$, then an su-dttr T exists such that $L(T) = L$.

By Propositions 1.4.7, 1.4.8 and 1.4.9, T_L can be computed from T and, by Lemma 2.2.4, it is an su-dttr, too. □

Theorem 2.2.6 *For any tree language $L \in DREC$ given by a dttr T recognizing L, it is decidable whether $L \in dom(sl\text{-}DT)$ holds.*

Proof. By Propositions 1.4.7, 1.4.8 and 1.4.9, T_L can be constructed effectively from T.

Moreover, it is obviously decidable whether T_L is an su-dttr. Hence, by Lemma 2.2.5 and Theorem 2.2.3, the statement of the theorem holds. □

Finally, we present an algorithm, which, for any tree language $L \in DREC$ given by a dttr recognizing L, decides whether $L \in$ dom($sl\text{-}DT$) holds. The method is based on the proof of the Theorem 2.2.6.

Let L be an arbitrary deterministic recognizable tree language and let $T^{(1)}$ be a dttr, which recognizes L.

The algorithm gives the answer *YES* if L can be the domain of a superlinear deterministic top-down tree transformation, otherwise it answers *NO*.

1. Compute $T^{(1)}_{nor}$ as defined in the proof of the Proposition 1.4.7. Denote $T^{(1)}_{nor}$ by $T^{(2)}$.

2. Compute $T^{(2)}_{con}$ as defined in the proof of the Proposition 1.4.8. Denote $T^{(2)}_{con}$ by $T^{(3)}$.

3. Construct $T^{(3)}_{min}$ as determined in Construction 1.4.10. Denote $T^{(3)}_{min}$ by T_L.

4. Decide whether T_L is semi-universal. (It is trivially decidable, e.g., check all rule pairs, which concern the same input symbol.) If it is, then the answer is *YES*, else the answer is *NO*.

2.3 Range tree languages

In this section we prove range(sl-DT) = REC. Furthermore, as a by-product, we get range(l-DT) = REC, too.

Assume that $L \in REC$, then there exists a ttr $T = (Q, \Sigma, \Sigma, q_0, R)$ satisfying $L(T) = L$. We define the ranked alphabet Δ such that, for each $m \geq 0$, we put

$$\Delta_m = \{\sigma_{q,q_1,\ldots,q_m} \mid \sigma \in \Sigma_m,\ q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \in R\}.$$

Let $T' = (Q, \Delta, \Sigma, q_0, R')$ be a dt tree transducer, where

$$R' = \{q(\sigma_{q,q_1,\ldots,q_m}(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \mid \sigma_{q,q_1,\ldots,q_m} \in \Delta\}.$$

Observe that T' is an rl-sl-dt tree transducer.

Lemma 2.3.1 *For any tree $t \in T_\Sigma$ and state $q \in Q$, $q(t) \Rightarrow_T^* t$ if and only if $q(t') \Rightarrow_{T'}^* t$ holds, for some $t' \in T_\Delta$.*

Proof. First assume $q(t) \Rightarrow_T^* t$. We show the existence of the above t' by induction on height(t).

Basis. Suppose that height(t) = 0, then $t = \delta$, for some $\delta \in \Sigma_0$, and then $q(t) \Rightarrow_T^* t$ implies $q(\delta) \to \delta \in R$. By the construction of T', $\delta_q \in \Delta_0$ and $q(\delta_q) \to \delta \in R'$, hence $q(t') \Rightarrow_{T'}^* t$ holds, for $t' = \delta_q$.

Induction step. Suppose that height(t) = $n + 1$ with $n \geq 0$, then $t = \sigma(t_1,\ldots,t_m)$, for some $m \geq 1$, $\sigma \in \Sigma_m$, and $t_1,\ldots,t_m \in T_\Sigma$, where height($t_i$) $\leq n$, for all $1 \leq i \leq m$. Since $q(t) \Rightarrow_T^* t$, there must be a rule $q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m))$ in R, where $q_i(t_i) \Rightarrow_T^* t_i$ holds, for each $1 \leq i \leq m$. By the construction of T', $\sigma_{q,q_1,\ldots,q_m} \in \Delta$ and the rule $q(\sigma_{q,q_1,\ldots,q_m}(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m))$ is in R. Moreover, by the induction hypothesis, there exist trees $t'_1,\ldots,t'_m \in T_\Delta$ such that $q_i(t'_i) \Rightarrow_{T'}^* t_i$, for all $1 \leq i \leq m$. Let $t' = \sigma_{q,q_1,\ldots,q_m}(t'_1,\ldots,t'_m)$, then we have $q(t') \Rightarrow_{T'} \sigma(q_1(t'_1),\ldots,q_m(t'_m)) \Rightarrow_{T'}^* \sigma(t_1,\ldots,t_m) = t$.

Now suppose that there exists a tree $t' \in T_\Delta$ satisfying $q(t') \Rightarrow_{T'}^* t$. We prove $q(t) \Rightarrow_T^* t$ also by induction on height(t). Recall that, since T' is relabeling, height(t') = height(t) necessarily holds.

Basis. Suppose that height(t) = 0. Then $t = \delta$, for some $\delta \in \Sigma_0$. By the construction of T', $t' = \delta_q \in \Delta_0$ and $q(\delta_q) \to \delta \in R'$, hence $q(\delta) \to \delta \in R$. Therefore, $q(t) \Rightarrow_T^* t$ holds.

Induction step. Let $n \geq 0$. Suppose that height$(t) = n+1$, then $t = \sigma(t_1, \ldots, t_m)$, for some $m \geq 1$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in T_\Sigma$, where height$(t_i) \leq n$ holds for all $1 \leq i \leq m$. By the construction of T', for some $q_1, \ldots, q_m \in Q$ and $t'_1, \ldots, t'_m \in T_\Delta$, $t' = \sigma_{q,q_1,\ldots,q_m}(t'_1, \ldots, t'_m)$ and $q(\sigma_{q,q_1,\ldots,q_m}(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ hold. Moreover $q_i(t'_i) \Rightarrow^*_{T'} t_i$, for each $1 \leq i \leq m$. Hence $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ and, by the induction hypothesis, $q_i(t_i) \Rightarrow^*_T t_i$, for all $1 \leq i \leq m$. Therefore, we have $q(t) \Rightarrow_T \sigma(q_1(t_1), \ldots, q_m(t_m)) \Rightarrow^*_T \sigma(t_1, \ldots, t_m) = t$. □

Lemma 2.3.1 implies that, for any tree $t \in T_\Sigma$, $q_0(t) \Rightarrow^*_T t$ holds if and only if there exists a tree $t' \in T_\Delta$ satisfying $q_0(t') \Rightarrow^*_{T'} t$. Hence $t \in L(T)$ if and only if $t \in$ range$(\tau_{T'})$. Since L was arbitrary and T' is rl-sl-dt tree transducer, it follows that $REC \subseteq$ range$(sl\text{-}DT)$.

On the other hand, range$(sl\text{-}DT) \subseteq$ range$(l\text{-}DT)$ obviously holds and, by Corollary 6.6 of Chapter IV in [GécSte4], range$(l\text{-}DT) \subseteq REC$, thus we have the following result.

Theorem 2.3.2 *range*$(sl\text{-}DT) =$ *range*$(l\text{-}DT) = REC$

Chapter 3

Hierarchy theorems of sl-dt tree transformations

It turned out in the previous chapter that, similarly to the classes DT and l-DT, the class sl-DT is not closed under the composition.

However, in Section 3.1 we show that, in contrast with the classes DT and l-DT, the hierarchy $\{sl\text{-}DT^n \mid n \geq 0\}$ never collapses. Moreover, we prove in Section 3.2 that even the hierarchy $\{t\text{-}sl\text{-}DT^n \mid n \geq 0\}$ is proper.

We note that the results of this chapter were published in [DánFül1].

3.1 The hierarchies dom(sl-DT^n) and sl-DT^n

When a tree transformation class is not closed under the composition, like sl-DT, it is always a fundamental question whether its increasing powers form an infinite hierarchy or not. In other words, whether its power hierarchy collapses at some integer or not.

An example for collapsing hierarchy is $\{DT^n \mid n \geq 1\}$. It was shown in [FülVág1] that $DT^2 = DT^3 = \ldots$.

On the other hand, there exists also proper hierarchies. For example, it is shown in [Eng1] that the hierarchy $\{NT^n \mid n \geq 1\}$ is proper, where NT denotes the class of nondeterministic top-down tree transformations.

In this section we show that the hierarchy $\{sl\text{-}DT^n \mid n \geq 1\}$ is proper. In fact, it will be the consequence of a stronger result, namely that the hierarchy of tree language classes $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 1\}$ is proper.

Thus first we prove that $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 1\}$ is proper. Our method is the following. For each $n \geq 2$, we define n sl-dt tree transducers $T^{1,n}, \ldots, T^{n,n}$. Then we show that, for every $s \geq 1$ and arbitrary sl-dt tree transducers $M_1, \ldots, M_s$, if $\mathrm{dom}(\tau_{T^{1,n}} \circ \cdots \circ \tau_{T^{n,n}}) = \mathrm{dom}(\tau_{M_1} \circ \cdots \circ \tau_{M_s})$, then $s \geq n$ necessarily holds.

Let $n \geq 2$ and define the ranked alphabets $\Sigma^{0,n}, \ldots, \Sigma^{n,n}$ as follows:

(i) $\Sigma^{0,n} = \{\#, \sigma_1, \ldots, \sigma_n\}$, where $\#$ has rank 0 and σ_j has rank 2, for each $1 \leq j \leq n$.

(ii) $\Sigma^{i,n} = \Sigma^{i-1,n} - \{\sigma_i\} = \{\#, \sigma_{i+1}, \ldots, \sigma_n\}$, for every $1 \leq i \leq n$.

Observe that $\Sigma^{n,n} = \{\#\}$.

Next, for every $1 \leq i \leq n$, define the sl-dt tree transducer

$$T^{i,n} = (Q, \Sigma^{i-1,n}, \Sigma^{i,n}, q, R^{i,n})$$

where

- $Q = \{q, q'\}$ and
- $R^{i,n}$ consists of the rules
  - $q(\sigma_i(x_1, x_2)) \to q'(x_1)$,
  - $q'(\sigma_i(x_1, x_2)) \to q(x_2)$,
  - $q(\sigma_j(x_1, x_2)) \to \sigma_j(q(x_1), q(x_2))$, for every j with $(i+1) \leq j \leq n$, and
  - $q(\#) \to \#$.

Informally speaking, the tree transducer $T^{i,n}$ works as follows. Its input alphabet is $\{\#, \sigma_i, \ldots, \sigma_n\}$. Given an input tree t, the tree transducer $T^{i,n}$, starting with its initial state q, performs an identical tree transformation on each σ_j with $i < j \leq n$.

When $T^{i,n}$ meets, in state q, a subtree t' of t with root σ_i, then it can process t' if and only if the following two conditions hold:

- $t' = \sigma_i(\sigma_i(t_1, t_2), t_3)$, for some input subtrees t_1, t_2, t_3.
- It can process t_2 starting in state q..

If this is the case, then the tree transducer $T^{i,n}$ deletes the two consecutive σ_i's together with the subtrees t_1 and t_3 from t' (and hence from t) and then processes t_2 with the state q.

We prove our theorem by inspecting how the sl-dt tree transducers $M_1, \ldots, M_s$ must work on some special trees. Therefore, for every $1 \leq i \leq n$, we define the trees $r_{1,i}, \ldots, r_{9,i}$ as can be seen in Figure 3.1.

For each $1 \leq i \leq n$, we introduce the notation $\tau_{i,n} = \tau_{T^{1,n}} \circ \ldots \circ \tau_{T^{i,n}}$. Moreover, we write τ_n for $\tau_{n,n}$ for brevity. It is an exercise to show that the following statement holds.

Statement 3.1.1 *Let $n \geq 2$ be an integer. Then the following conditions hold:*

- *For every i such that $1 \leq i \leq n$,*

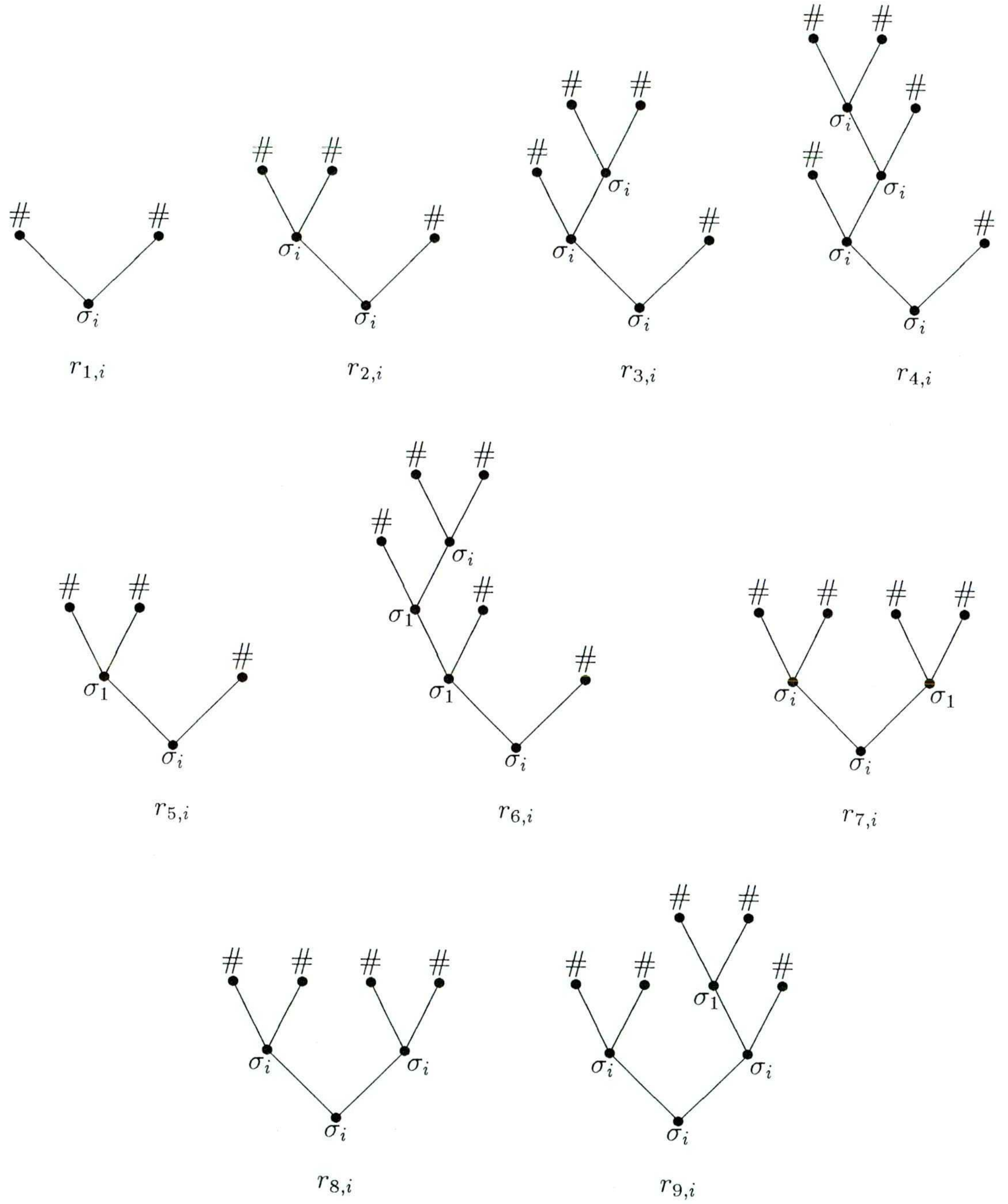

Figure 3.1: Example trees for hierarchy theorems

- $r_{1,i}, r_{3,i} \notin \mathrm{dom}(\tau_n)$ *and*
- $r_{2,i}, r_{4,i} \in \mathrm{dom}(\tau_n)$.

- *For every i such that $2 \leq i \leq n$,*
  - $r_{5,i}, r_{7,i}, r_{9,i} \notin \mathrm{dom}(\tau_n)$ *and*
  - $r_{6,i}, r_{8,i} \in \mathrm{dom}(\tau_n)$.

Now we are ready to prove the following lemma, which serves as the foundation of the results of this section.

Lemma 3.1.2 *Let $n \geq 2$ and $s \geq 1$. Moreover, let $M_1, \ldots, M_s$ be an arbitrary sequence of sl-dt tree transducers and put $\mu_s = \tau_{M_1} \circ \cdots \circ \tau_{M_s}$. Then $dom(\tau_n)$=$dom(\mu_s)$ implies $s \geq n$.*

Proof. We prove the lemma by induction on n.

The basis $n = 2$. We prove by contradiction that $\mathrm{dom}(\tau_2)$ cannot be the domain of any sl-dt tree transformation. Therefore, suppose that there exists an sl-dt tree transducer

$$T = (Q', \Sigma^{0,2}, \Delta, p, R)$$

such that $\mathrm{dom}(\tau_T)$ =$\mathrm{dom}(\tau_2)$.

We shall investigate the rules in R. Since $\# \in \mathrm{dom}(\tau_2)$, there must be a $(p, \#)$-rule in R, hence

$$p(\#) \to t_\# \in R,$$

for some $t_\# \in T_\Delta$.

By Statement 3.1.1, $r_{2,2}$ is in $\mathrm{dom}(\tau_2)$. Therefore there must be a (p, σ_2)-rule in R. Moreover, since T is linear, that (p, σ_2)-rule must be of one of the forms specified in (1)–(5). We shall show that each case leads to a contradiction.

(1) Assume that $p(\sigma_2(x_1, x_2)) \to t \in R$, where $t \in T_\Delta$ is a ground term. This implies $r_{1,2}$ $\in$$\mathrm{dom}(\tau_T)$, which contradicts Statement 3.1.1.

(2) Assume that $p(\sigma_2(x_1, x_2)) \to t[p'(x_1), p''(x_2)] \in R$, where $p', p'' \in Q'$ and $t \in \hat{T}_{\Delta,2}$. Since $r_{1,2} \notin \mathrm{dom}(\tau_2)$, there cannot be $(p', \#)$ and $(p'', \#)$-rules in R simultaneously. On the other hand, $r_{2,2}$ is in $\mathrm{dom}(\tau_2)$, thus a $(p'', \#)$-rule should be in R. Hence we can conclude that there is no $(p', \#)$-rule in R. Moreover, $r_{2,2} \in \mathrm{dom}(\tau_2)$ implies that there must be a (p', σ_2)-rule in R. Let this rule be specified as

$$p'(\sigma_2(x_1, x_2)) \to t',$$

where $t' \in T_\Delta(Q(X_2))$. Observe that the tree t' must contain the variable x_2, otherwise $r_{2,2} \in \mathrm{dom}(\tau_T)$ implies $r_{3,2} \in \mathrm{dom}(\tau_T)$, which contradicts

Statement 3.1.1. Moreover, T is superlinear and both x_1 and x_2 occur in $\mathrm{rhs}(p, \sigma_2)$. Therefore a variable can occur in $t' = \mathrm{rhs}(p', \sigma_2)$ if and only if $p' = p$. However, this is impossible, because there is a $(p, \#)$-rule in R, as we saw above.

(3) Suppose that $p(\sigma_2(x_1, x_2)) \to t[p'(x_2), p''(x_1)] \in R$, where $p', p'' \in Q'$ and $t \in \hat{T}_{\Delta,2}$. Observe that this case is quite similar to the previous one. The only difference is that the roles of p' and p'' are interchanged, hence such a rule in R again implies a contradiction.

(4) Assume that $p(\sigma_2(x_1, x_2)) \to t[p'(x_2)] \in R$, where $p' \in Q'$ and $t \in \hat{T}_{\Delta,1}$. In this case, since the first subtree of σ_2 is deleted, the condition $r_{2,2} \in \mathrm{dom}(\tau_T)$ holds if and only if $r_{1,2} \in \mathrm{dom}(\tau_T)$, which contradicts Statement 3.1.1.

(5) Suppose that the rule $p(\sigma_2(x_1, x_2)) \to t[p'(x_1)]$ is in R, where $p' \in Q'$ and $t \in \hat{T}_{\Delta,1}$. Since the second subtree of σ_2 is deleted, we get $r_{2,2} \in \mathrm{dom}(\tau_T)$ if and only if $r_{7,2} \in \mathrm{dom}(\tau_T)$. This is contradiction, by Statement 3.1.1.

We have shown that no sl-dt tree transducer T exists such that $\mathrm{dom}(\tau_T) = \mathrm{dom}(\tau_2)$. Hence our lemma is proved for $n = 2$.

Induction step. Suppose the lemma has been proved for $n-1$. Moreover, assume the sl-dt tree transducers $M_1, \ldots, M_s$ to be such that $\mathrm{dom}(\tau_n) = \mathrm{dom}(\mu_s)$.

By (4) of Corollary 2.1.12, an op-ni-sl-dt tree transducer $M_{1,1}$ and an nr-l-nd-hom tree transducer $M_{1,2}$ exist such that $\tau_{M_1} = \tau_{M_{1,1}} \circ \tau_{M_{1,2}}$. Moreover, by Corollary 2.1.2 and by (3) of Corollary 2.1.12, the tree transducer $M_2' = M_{1,2} \circ M_2$ is superlinear and $\tau_{M_2'} = \tau_{M_{1,2}} \circ \tau_{M_2}$ holds.

Hence we have

$$\begin{aligned} \mu_s &= \tau_{M_{1,1}} \circ \tau_{M_{1,2}} \circ \tau_{M_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s} \\ &= \tau_{M_{1,1}} \circ \tau_{M_2'} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}. \end{aligned}$$

Roughly speaking, we "push forward" the undesirable properties, namely the variable permuting and the height increasing, by specializing the first tree transducer.

Suppose that

$$M_{1,1} = (Q', \Sigma^{0,n}, \Omega, p, R).$$

Similarly to the case $n = 2$, we investigate the rules of R. Since $\# \in \mathrm{dom}(\tau_n)$, there must be a $(p, \#)$-rule in R. Moreover, since $M_{1,1}$ is nonincreasing, this rule should be of the form

$$p(\#) \to \#^p,$$

where $\#^p \in \Omega_0$.

Now let the integer i be arbitrary but satisfying $2 \le i \le n$. Since $r_{2,i} \in \mathrm{dom}(\tau_n)$, there must be a (p, σ_i)-rule in R. Moreover, $M_{1,1}$ is order preserving, nonincreasing and linear. Hence the (p, σ_i)-rule must be of one of the forms (1)–(4) detailed below.

(1) Assume that $p(\sigma_i(x_1, x_2)) \to \sigma_i^p \in R$, where $\sigma_i^p \in \Omega_0$. In this case, since the subtrees of σ_i are deleted, $r_{2,i} \in \mathrm{dom}(\mu_s)$ if and only if $r_{1,i} \in \mathrm{dom}(\mu_s)$, which contradicts Statement 3.1.1. (Note that we have assumed $\mathrm{dom}(\tau_n) = \mathrm{dom}(\mu_s)$.)

(2) Suppose that $p(\sigma_i(x_1, x_2)) \to t_i[p_i(x_1)] \in R$, where either $t_i = x_1$ or $t_i = \sigma_i^p(x_1)$ holds for some $\sigma_i^p \in \Omega_1$ and $p_i \in Q'$. The rule deletes the second subtree of σ_i, and therefore $r_{2,i} \in \mathrm{dom}(\mu_s)$ if and only if $r_{7,i} \in \mathrm{dom}(\mu_s)$, contradicting Statement 3.1.1.

(3) Assume that $p(\sigma_i(x_1, x_2)) \to t_i[p_i(x_2)] \in R$, where either $t_i = x_1$ or $t_i = \sigma_i^p(x_1)$, for some $\sigma_i^p \in \Omega_1$ and $p_i \in Q'$. Since the rule deletes the first subtree of σ_i, $r_{2,i} \in \mathrm{dom}(\mu_s)$ if and only if $r_{1,i} \in \mathrm{dom}(\mu_s)$, which contradicts Statement 3.1.1.

(4) We have obtained the fact that the only possible form for the (p, σ_i)-rule is

$$p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p_i(x_1), p_i'(x_2)),$$

where $\sigma_i^p \in \Omega_2$ and $p_i, p_i' \in Q'$.

Since $r_{8,i} \in \mathrm{dom}(\tau_n)$, both a (p_i, σ_i)-rule and a (p_i', σ_i)-rule must be in R. If $\mathrm{rhs}(p_i, \sigma_i)$ were a ground tree, then $r_{2,i} \in \mathrm{dom}(\mu_s)$ would imply $r_{3,i} \in \mathrm{dom}(\mu_s)$, which would contradict Statement 3.1.1.

Similarly, $\mathrm{rhs}(p_i', \sigma_i)$ cannot be a ground tree. Indeed, if it were, then $r_{8,i} \in \mathrm{dom}(\mu_s)$ would imply $r_{9,i} \in \mathrm{dom}(\mu_s)$, contradicting Statement 3.1.1.

It follows that both $\mathrm{rhs}(p_i, \sigma_i)$ and $\mathrm{rhs}(p_i', \sigma_i)$ must contain a variable. However, since $M_{1,1}$ is superlinear, this is possible if and only if $p_i = p_i' = p$. Hence we have shown that

$$p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p(x_1), p(x_2)) \in R.$$

Recall that also $p(\#) \to \#^p \in R$.

We can suppose without loss of generality that the symbols $\#, \sigma_2, \ldots, \sigma_n$ are not in Ω, which is the output ranked alphabet of $M_{1,1}$ and the input ranked alphabet of M_2'. (Otherwise, we can easily relabel them both in $M_{1,1}$ and M_2' in such a way that the induced tree transformation $\tau_{M_{1,1}} \circ \tau_{M_2'}$ remains the same.)

Then write $\#, \sigma_2, \ldots, \sigma_n$ for $\#^p, \sigma_2^p, \ldots, \sigma_n^p$, respectively, in Ω and both in the right-hand sides of rules of $M_{1,1}$ and in the left-hand sides of rules of M_2'. We denote the resulting tree transducers by $\overline{M}_1$ and $\overline{M}_2$, respectively. It should be obvious that

$$\tau_{M_{1,1}} \circ \tau_{M_2'} = \tau_{\overline{M}_1} \circ \tau_{\overline{M}_2}$$

and, by our results on the rules in R, $\tau_{\overline{M}_1}|_{T_{\Sigma^{1,n}}} = \mathrm{id}(T_{\Sigma^{1,n}})$. Consequently,

$$\begin{aligned}
\mathrm{dom}(\mu_s|_{T_{\Sigma^{1,n}}}) &= \mathrm{dom}(\tau_{M_{1,1}} \circ \tau_{M_2'} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}|_{T_{\Sigma^{1,n}}}) \\
&= \mathrm{dom}(\tau_{\overline{M}_1} \circ \tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}|_{T_{\Sigma^{1,n}}}) \\
&= \mathrm{dom}(\tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}|_{T_{\Sigma^{1,n}}}).
\end{aligned}$$

Let the nr-l-hom tree transducer T be such that $\tau_T = \mathrm{id}(T_{\Sigma^{1,n}})$ holds. Obviously, there exists such a T, because $I \subseteq nr\text{-}l\text{-}HOM$ (see Subsection 1.4.1).

Consider the dt tree transducer N_2 defined by $N_2 = T \circ \overline{M}_2$. Note that T is total, superlinear, and nonreducing. Therefore, by Corollary 2.1.2 and by (3) of Corollary 2.1.12, N_2 is superlinear and $\tau_{N_2} = \tau_T \circ \tau_{\overline{M}_2}$ holds. Hence we can compute on as follows:

$$\begin{aligned}
\mathrm{dom}(\tau_{\overline{M}_2} \circ \tau_{M_3} \circ \ldots \circ \tau_{M_s}|_{T_{\Sigma^{1,n}}}) &= \mathrm{dom}(\tau_T \circ \tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}) \\
&= \mathrm{dom}(\tau_{N_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}).
\end{aligned}$$

On the other hand, observe that $T^{1,n}$ and $T^{2,n}$ are defined so that $\tau_{T^{1,n}}|_{T_{\Sigma^{1,n}}} = \mathrm{id}(T_{\Sigma^{1,n}})$ and $\mathrm{dom}(\tau_{T^{2,n}}) \subseteq T_{\Sigma^{1,n}}$. This implies

$$\begin{aligned}
\mathrm{dom}(\tau_n|_{T_{\Sigma^{1,n}}}) &= \mathrm{dom}(\tau_{T^{1,n}} \circ \cdots \circ \tau_{T^{n,n}}|_{T_{\Sigma^{1,n}}}) \\
&= \mathrm{dom}(\tau_{T^{2,n}} \circ \cdots \circ \tau_{T^{n,n}}).
\end{aligned}$$

Hence, by $\mathrm{dom}(\mu_s|_{T_{\Sigma^{1,n}}}) = \mathrm{dom}(\tau_n|_{T_{\Sigma^{1,n}}})$, we have

$$\mathrm{dom}(\tau_{N_2} \circ \tau_{M_3} \circ \ldots \circ \tau_{M_s}) = \mathrm{dom}(\tau_{T^{2,n}} \circ \ldots \circ \tau_{T^{n,n}}). \tag{$*$}$$

Now we would like to apply our induction hypothesis to $(*)$. However, we cannot do that in the present form of $(*)$, because $T^{2,n}, \ldots, T^{n,n}$ appear in it instead of $T^{1,n-1}, \ldots, T^{n-1,n-1}$. Fortunately, we can get the required form by suitably relabeling in $T^{2,n}, \ldots, T^{n,n}$.

We can observe the following. For every $2 \le i \le n$, if we write $\sigma_{i-1}, \ldots, \sigma_{n-1}$ for $\sigma_i, \ldots, \sigma_n$, respectively, in the specification of $T^{i,n}$, then we get $T^{i-1,n-1}$. Hence $(*)$ implies

$$\mathrm{dom}(\tau_{N_2'} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}) = \mathrm{dom}(\tau_{T^{1,n-1}} \circ \cdots \circ \tau_{T^{n-1,n-1}}),$$

where N_2' is obtained from N_2 by writing $\sigma_1, \ldots, \sigma_{n-1}$ for $\sigma_2, \ldots, \sigma_n$ in the input alphabet and in the left-hand sides of the rules of N_2.

We obtained that $\mathrm{dom}(\tau_{n-1})$ appears as the domain of the composition of $s-1$ sl-dt tree transformations. Then, the by induction hypothesis, $s-1 \ge n-1$, which implies $s \ge n$.

With this we finished the proof of the lemma. □

On the basis of Lemma 3.1.2 we can easily prove the following important results.

Theorem 3.1.3 *For any integer $n \geq 1$, the following inclusions hold:*

(1) $dom(sl\text{-}DT^n) \subset dom(sl\text{-}DT^{n+1})$.
(2) $dom(sl\text{-}DT^n) \subset DREC$.
(3) $sl\text{-}DT^n \subset sl\text{-}DT^{n+1}$.

Proof.
(1) Obviously, $\mathrm{dom}(sl\text{-}DT^n) \subseteq \mathrm{dom}(sl\text{-}DT^{n+1})$. On the other hand, by Lemma 3.1.2, $\mathrm{dom}(\tau_{n+1}) \notin \mathrm{dom}(sl\text{-}DT^n)$.

(2) We recall from Section 1.4.5 that $\mathrm{dom}(l\text{-}DT^n) = DREC$ holds, for every $n \geq 1$. Hence we have

$$\mathrm{dom}(sl\text{-}DT^n) \subset \mathrm{dom}(sl\text{-}DT^{n+1}) \subseteq \mathrm{dom}(l\text{-}DT^{n+1}) = \mathrm{dom}(l\text{-}DT) = DREC.$$

(3) It should be clear, by the proof of (1), that $\tau_{n+1} \notin sl\text{-}DT^n$. $\square$

Finally we present the inclusion diagram of the classes DT^2, DT, $l\text{-}DT^2$, $l\text{-}DT$, and $sl\text{-}DT^n$ with $n \geq 1$ in Figure 3.2. The proper inclusions shown by the diagram follow from Theorems 2.1.4, 2.1.13, 3.1.3 and Corollary 2.1.14.

3.2 The hierarchy $t\text{-}sl\text{-}DT^n$

Total tree transducers are of special interest in the theory of tree transformations. Namely, it is known from [Rou] that although the classes DT and $l\text{-}DT$ are not closed under the composition, the subclasses $t\text{-}DT$ and $t\text{-}l\text{-}DT$ of the corresponding total tree transformations are (see also Proposition 1.4.1). This explains our motivation to study the class $t\text{-}sl\text{-}DT$.

In the previous section we proved that $\{sl\text{-}DT^n \mid n \geq 1\}$ forms a proper hierarchy, which implies that $sl\text{-}DT$ is not closed under the composition. In this section we show that even the hierarchy $\{t\text{-}sl\text{-}DT^n \mid n \geq 1\}$ is proper.

We follow the method that we applied in the previous section. Namely, for each $n \geq 2$, we define n t-sl-dt tree transducers $T^{1,n}, \ldots, T^{n,n}$. Then we show that, for every $s \geq 1$ and arbitrary t-sl-dt tree transducers $M_1, \ldots, M_s$, if $\tau_{T^{1,n}} \circ \cdots \circ \tau_{T^{n,n}} = \tau_{M_1} \circ \cdots \circ \tau_{M_s}$, then $s \geq n$.

Therefore, let $n \geq 2$ be an arbitrary integer. Define the ranked alphabets $\Sigma^{0,n}, \ldots, \Sigma^{n,n}$ as follows:

(i) $\Sigma^{0,n} = \{\#, \$, \sigma_1, \ldots, \sigma_n\}$ where $\#$ and $\$$ have rank 0 and σ_j has rank 2, for each $1 \leq j \leq n$.

(ii) $\Sigma^{i,n} = \Sigma^{i-1,n} - \{\sigma_i\} = \{\#, \$, \sigma_{i+1}, \ldots, \sigma_n\}$, for every $1 \leq i \leq n$.

It should be clear that $\Sigma^{n,n} = \{\#, \$\}$.

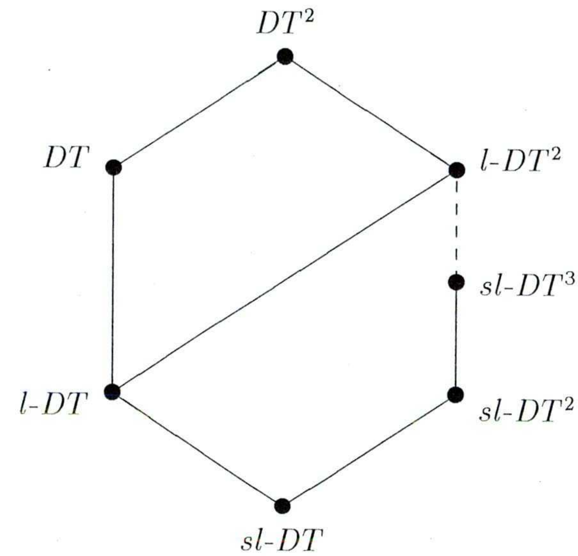

Figure 3.2: The hierarchy of $sl\text{-}DT^n$

Next, for every $1 \leq i \leq n$, define the t-sl-dt tree transducer

$$T^{i,n} = (Q, \Sigma^{i-1,n}, \Sigma^{i,n}, q, R^{i,n}),$$

where

- $Q = \{q, q'\}$ and
- $R^{i,n}$ consists of the rules
  - $q(\sigma_i(x_1, x_2)) \to q'(x_1)$,
  - $q'(\sigma_i(x_1, x_2)) \to q(x_2)$,
  - $q(\sigma_j(x_1, x_2)) \to \sigma_j(q(x_1), q(x_2))$, for every $(i+1) \leq j \leq n$,
  - $q'(\sigma_j(x_1, x_2)) \to \$$, for every $(i+1) \leq j \leq n$, and
  - $q(\#) \to \#$, $q(\$) \to \$$, $q'(\#) \to \$$, $q'(\$) \to \$$.

Roughly speaking, the t-sl-dt tree transducer $T^{i,n}$ works as follows. Its input alphabet is $\{\#, \$, \sigma_i, \ldots, \sigma_n\}$. Given an input tree t, the tree transducer $T^{i,n}$, starting with its initial state q, performs an identical tree transformation for each σ_j with $i < j \leq n$. When $T^{i,n}$ meets, in state q, a subtree t' of t with root σ_i, then two cases are possible.

Case 1: $t' = \sigma_i(\sigma_i(t_1, t_2), t_3)$, for some input subtrees t_1, t_2, t_3. If this is the case, then $T^{i,n}$ deletes the two σ_i's together with the subtrees t_1 and t_3 from t' (and hence from t) and then processes t_2 with state q.

Case 2: t' has some other form (although its root is σ_i). Then $T^{i,n}$ transforms the tree t' to \$.

Again, we introduce the notation $\tau_{i,n} = \tau_{T^{1,n}} \circ \cdots \circ \tau_{T^{i,n}}$, for every $1 \leq i \leq n$. We write τ_n for $\tau_{n,n}$. Moreover, we again refer to the trees depicted in Figure 3.1.
The proof of the following statement is an easy exercise.

Statement 3.2.1 *Let $n \geq 2$ be an integer. Then the following statements hold:*

- *For every i such that $1 \leq i \leq n$,*
  - $\tau_n(r_{1,i}) = \tau_n(r_{3,i}) = \$$ *and*
  - $\tau_n(r_{2,i}) = \tau_n(r_{4,i}) = \#$.
- *For every i such that $2 \leq i \leq n$,*
  - $\tau_n(r_{5,i}) = \$$ *and*
  - $\tau_n(r_{6,i}) = \tau_n(r_{7,i}) = \tau_n(r_{8,i}) = \tau_n(r_{9,i}) = \#$.

We can observe the following. For every $1 \leq i \leq n$, $\tau_{i,n}(t) = \$$ implies that t is not in the domain of the partial tree transformation corresponding to $\tau_{i,n}$, c.f. the previous section.

However, the converse is not true. For example, $\tau_n(r_{7,i}) = \#$ holds, despite $r_{7,i}$ being not in the domain of the partial tree transformation corresponding to τ_n, see Statement 3.1.1.

Our key lemma now sounds as follows.

Lemma 3.2.2 *Let $n \geq 2$ and $s \geq 1$. Moreover, let $M_1, \ldots, M_s$ be arbitrary t-sl-dt tree transducers and put $\mu_s = \tau_{M_1} \circ \cdots \circ \tau_{M_s}$. Then $\tau_n = \mu_s$, implies $s \geq n$.*

Proof. We prove also this lemma by induction on n.

The basis $n = 2$. We prove by contradiction that τ_2 cannot be induced by any t-sl-dt tree transducer. Therefore, assume that there is a t-sl-dt tree transducer

$$T = (Q', \Sigma^{0,2}, \Sigma^{2,2}, p, R)$$

such that $\tau_T = \tau_2$ holds.

Note that both the input and the output ranked alphabets of T are determined by the condition $n = 2$.

We investigate the rules of R. Since $\tau_2(\#) = \#$ and $\tau_2(\$) = \$$, the rules

$$p(\#) \to \#$$

and

$$p(\$) \to \$$$

must be in R.

Let $1 \leq i \leq 2$ and consider the (p, σ_i)-rule. Since the output ranked alphabet of T contains symbols having rank 0 only, the (p, σ_i)-rule must be of one of the forms specified in (1)–(4):

(1) $p(\sigma_i(x_1, x_2)) \to \#$,

(2) $p(\sigma_i(x_1, x_2)) \to \$$,

(3) $p(\sigma_i(x_1, x_2)) \to p_i(x_1)$,

(4) $p(\sigma_i(x_1, x_2)) \to p_i(x_2)$,

where $p_i \in Q'$. Observe that, in cases (1), (2) and (4) the application of the (p, σ_i)-rule deletes the first subtree of σ_i. Therefore, in these cases, $\tau_T(r_{2,i}) = \tau_T(r_{1,i})$ holds, which contradicts Statement 3.2.1. Hence we obtain that the only possible form is (3), that is to say

$$p(\sigma_i(x_1, x_2)) \to p_i(x_1) \in R.$$

Since $\tau_2(r_{1,i}) = \$$, we have

$$p_i(\#) \to \$ \in R.$$

Thus the states p and p_i must be different.

Now consider the (p_i, σ_i)-rule. Suppose that $\mathrm{rhs}(p_i, \sigma_i)$ is a ground tree. Then it must be $\#$, since $\tau_T(r_{2,i}) = \#$. This, however, implies $\tau_T(r_{3,i}) = \#$, which contradicts Statement 3.2.1.

We obtain that $\mathrm{rhs}(p_i, \sigma_i)$ must contain a variable. Noticing that T is superlinear and that x_1 already appears in $\mathrm{rhs}(p, \sigma_i)$, we get that the (p_i, σ_i)-rule must be of the form

$$p_i(\sigma_i(x_1, x_2)) \to p_i'(x_2),$$

for some $p_i' \in Q'$. Since $\tau_2(r_{2,i}) = \#$,

$$p_i'(\#) \to \# \in R$$

must hold, and therefore $p_i \neq p_i'$.

Consider the (p_i', σ_i)-rule. If $\mathrm{rhs}(p_i', \sigma_i)$ were a ground tree, then it must be $\#$, because $\tau_T(r_{4,i}) = \#$. Then $\tau_T(r_{3,i}) = \#$ would follow, contradicting Statement 3.2.1.

Therefore, $\mathrm{rhs}(p_i', \sigma_i)$ must also contain a variable. However T is superlinear, hence $\mathrm{rhs}(p_i', \sigma_i)$ can contain a variable if and only if $p_i' = p$.

In summary, the rules

$$p(\#) \to \#,$$

$$p(\$) \to \$,$$

$$p_i(\#) \to \$,$$

$$p(\sigma_i(x_1, x_2)) \to p_i(x_1),$$

and

$$p_i(\sigma_i(x_1, x_2)) \to p(x_2)$$

must be in R, for $i = 1, 2$. Moreover, $p \neq p_i$ holds.

Now consider the (p_2, σ_1)-rule. We observe that $\mathrm{rhs}(p_2, \sigma_1)$ must contain the variable x_1, otherwise $\tau_T(r_{5,2}) = \tau_T(r_{6,2})$ contradicts Statement 3.2.1. On the other hand, $\mathrm{rhs}(p, \sigma_1)$ also contains x_1. Since T is superlinear, we get $p = p_2$, which is a contradiction.

With this we proved that a suitable t-sl-dt tree transducer T does not exist.

The induction step. Suppose the lemma has been proved for $n-1$. Moreover, assume the t-sl-dt tree transducers $M_1, \ldots, M_s$ to be such that $\mu_s = \tau_n$.

By (5) of Corollary 2.1.12, a t-op-ni-sl-dt tree transducer $M_{1,1}$ and a nr-l-nd-hom tree transducer $M_{1,2}$ exist such that $\tau_{M_1} = \tau_{M_{1,1}} \circ \tau_{M_{1,2}}$.

Define the tree transducer $M_2' = M_{1,2} \circ M_2$. Observe that, by Lemma 2.1.11 and by (1) and (3) of Proposition 1.4.1, M_2' is t-sl-dt and $\tau_{M_2'} = \tau_{M_{1,2}} \circ \tau_{M_2}$.

Hence we have the equations

$$\begin{aligned} \mu_s &= \tau_{M_1} \circ \tau_{M_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s} \\ &= \tau_{M_{1,1}} \circ \tau_{M_2'} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}. \end{aligned}$$

We specify $M_{1,1}$ as

$$M_{1,1} = (Q', \Sigma^{0,n}, \Omega, p, R).$$

Let us examine the rules in R. Since $M_{1,1}$ is nonincreasing, the rules

$$p(\#) \to \#^p$$

and

$$p(\$) \to \$^p$$

are in R, for some $\#^p, \$^p \in \Omega_0$.

Now let i be arbitrary such that $1 \le i \le n$. Consider the (p, σ_i)-rule. We observe that, since $M_{1,1}$ is linear, order preserving and nonincreasing, it must be of one of the following forms:

(1) $p(\sigma_i(x_1, x_2)) \to \sigma_i^p$, where $\sigma_i^p \in \Omega_0$.

(2) $p(\sigma_i(x_1, x_2)) \to t_i[p_i(x_1)]$, where $p_i \in Q'$ and either $t_i = x_1$ or $t_i = \sigma_i^p(x_1)$, for some $\sigma_i^p \in \Omega_1$.

(3) $p(\sigma_i(x_1, x_2)) \to t_i[p_i(x_2)]$, where $p_i \in Q'$ and either $t_i = x_1$ or $t_i = \sigma_i^p(x_1)$, for some $\sigma_i^p \in \Omega_1$.

(4) $p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p_i(x_1), p_i'(x_2))$, where $p_i, p_i' \in Q'$ and $\sigma_i^p \in \Omega_2$.

We show that cases (1)–(3) lead to a contradiction.

Indeed, in cases (1) and (3), the first subtree of σ_i is deleted. Hence, in both cases, $\mu_s(r_{1,i}) = \mu_s(r_{2,i})$ holds, which contradicts Statement 3.2.1. (Note that we assumed $\mu_s = \tau_n$.)

Next, suppose that the case (2) holds. Consider the (p_i, σ_i)-rule. We observe that $\text{rhs}(p_i, \sigma_i)$ must contain x_2. For if it does not, then $\mu_s(r_{2,i}) = \mu_s(r_{3,i})$ holds, which contradicts Statement 3.2.1. This implies that $p \ne p_i$.

Since $M_{1,1}$ is a superlinear and nonincreasing tree transducer, the (p_i, σ_i)-rule must be of the form

$$p_i(\sigma_i(x_1, x_2)) \to t_i'[p_i'(x_2)],$$

where $p_i' \in Q'$ and either $t_i' = x_1$ or $t_i' = \sigma_i^{p_i}$, for some $\sigma_i^{p_i} \in \Omega_1$. The (p_i', σ_i)-rule must contain x_1 on its right-hand side, otherwise $\mu_s(r_{3,i}) = \mu_s(r_{4,i})$ holds, contradicting Statement 3.2.1. However, since $M_{1,1}$ is superlinear, this is possible if and only if $p_i' = p$.

From now on, let $2 \leq i \leq n$ and consider the (p_i, σ_1)-rule. The tree $\mathrm{rhs}(p_i, \sigma_1)$ must contain x_1, otherwise $\mu_s(r_{5,i}) = \mu_s(r_{6,i})$ contradicts Statement 3.2.1. On the other hand, the (p, σ_1)-rule also contains x_1 on its right-hand side. Since the tree transducer $M_{1,1}$ is superlinear, it follows that $p_i = p$, for every $2 \leq i \leq n$, which is a contradiction.

We got that the (p, σ_i)-rule cannot delete any of the subtrees of σ_i. Therefore its only possible form is (4), that is to say,

$$p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p_i(x_1), p_i'(x_2)) \in R,$$

where $p_i, p_i' \in Q'$ and $\sigma_i^p \in \Omega_2$. We show that $p_i = p_i' = p$.

Really, $M_{1,1}$ is superlinear, hence if $p \neq p_i$, then $\mathrm{rhs}(p_i, \sigma_i)$ must be a ground tree, which implies the contradiction $\mu_s(r_{2,i}) = \mu_s(r_{3,i})$. Consequently $p = p_i$, that is to say,

$$p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p(x_1), p_i'(x_2)) \in R.$$

However, in this case the assumption $p \neq p_i'$ leads to the contradiction $\mu_s(r_{3,i}) = \mu_s(r_{4,i})$, hence $p_i' = p$ holds, too.

We have obtained that the rules

$$p(\#) \to \#^p,$$

$$p(\$) \to \$^p,$$

and

$$p(\sigma_i(x_1, x_2)) \to \sigma_i^p(p(x_1), p(x_2))$$

are in R.

Similarly as in the proof of Lemma 3.1.2, we can suppose without loss of generality that the symbols $\#, \$, \sigma_2, \ldots, \sigma_n$ do not occur in Ω, which is the output alphabet of $M_{1,1}$ and the input alphabet of M_2'.

Then write $\#, \$, \sigma_2, \ldots, \sigma_n$ for $\#^p, \$^p, \sigma_2^p, \ldots, \sigma_n^p$, respectively, in Ω and both in the right-hand sides of rules of $M_{1,1}$ and in the left-hand sides of rules of M_2'. We denote the obtained t-sl-dt tree transducers by $\overline{M}_1$ and $\overline{M}_2$, respectively. Clearly, $\tau_{M_{1,1}} \circ \tau_{M_2'} = \tau_{\overline{M}_1} \circ \tau_{\overline{M}_2}$ holds.

What we have shown about R implies that the rules

$$p(\#) \to \#,$$

$$p(\$) \to \$,$$

and

$$p(\sigma_i(x_1, x_2)) \to \sigma_i(p(x_1), p(x_2))$$

will belong to $\overline{M}_1$. Hence $\overline{M}_1$ induces the identity tree transformation on $T_{\Sigma^{1,n}}$, that is to say, $\tau_{\overline{M}_1}|_{T_{\Sigma^{1,n}}} = \mathrm{id}(T_{\Sigma^{1,n}})$.

From now on, the proof is very similar to that of Lemma 3.1.2, hence we omit the details.

Let T be an nr-l-hom tree transducer such that $\tau_T = \mathrm{id}(T_{\Sigma^{1,n}})$. Then we have

$$\mu_s|_{T_{\Sigma^{1,n}}} = \tau_T \circ \tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}.$$

Now, let $N_2 = T \circ \overline{M}_2$. Then N_2 is total, superlinear, and

$$\tau_T \circ \tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s} = \tau_{N_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s}.$$

On the other hand,

$$\tau_n|_{T_{\Sigma^{1,n}}} = \tau_{T^{2,n}} \circ \cdots \circ \tau_{T^{n,n}}.$$

Thus we get

$$\tau_{\overline{M}_2} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s} = \tau_{T^{2,n}} \circ \cdots \circ \tau_{T^{n,n}}.$$

Now we are able to apply the induction hypothesis. By a suitable relabeling, we obtain

$$\tau_{N_2'} \circ \tau_{M_3} \circ \cdots \circ \tau_{M_s} = \tau_{T^{1,n-1}} \circ \cdots \circ \tau_{T^{n-1,n-1}}.$$

We have shown that τ_{n-1} appears as the composition of $s-1$ t-sl-dt tree transformations. Then, by induction hypothesis, $s-1 \geq n-1$, yielding $s \geq n$.

With this we finished the proof of the lemma. □

Using Lemma 3.2.2 it is easy to prove the main result of this section, which sounds as follows.

Theorem 3.2.3 *For any integer $n \geq 1$,*

$$t\text{-}sl\text{-}DT^n \subset t\text{-}sl\text{-}DT^{n+1}$$

holds.

Proof. It should be clear that $t\text{-}sl\text{-}DT^n \subseteq t\text{-}sl\text{-}DT^{n+1}$. Moreover, by Lemma 3.2.2, we have $\tau_{n+1} \notin t\text{-}sl\text{-}DT^n$. □

We present the inclusion diagram of the tree transformation classes *t-DT*, *t-l-DT*, and *t-sl-DT*^{n} with $n \geq 1$ in Figure 3.3. The proper inclusions shown by the diagram follow from Corollary 2.1.14 and Theorem 3.2.3. Note that this is in fact the total version of the diagram in Figure 3.2.

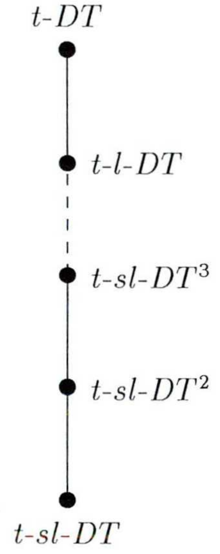

Figure 3.3: The hierarchy of t-sl-DT^n

Chapter 4

Compositions with sl-dt tree transformations

In this chapter we characterize the composition tree transformation classes of the form $X_1 \circ \ldots \circ X_n$, where $n \geq 0$ and, for each $1 \leq i \leq n$, X_i is the element of the set $M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}$. Namely, for arbitrary composition classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of the above form, we want to know whether $\mathcal{C}_1 \subseteq \mathcal{C}_2$, $\mathcal{C}_1 \supseteq \mathcal{C}_2$, $\mathcal{C}_1 = \mathcal{C}_2$, or $\mathcal{C}_1 \bowtie \mathcal{C}_2$ holds.

Similar question was already investigated and a general method was proposed for solving these kind of problems in [FülVág4]. A description of that general method is also presented in [FülVág6]. In the works [FülVág2], [FülVág4], and [FülVág5] the general method was implemented for a set consisting of DT and six of its subclasses. Moreover, the method was also applied to a set consisting of deterministic bottom-up tree transformation classes in [Fül2], for a set of deterministic top-down tree transformation classes with regular look-ahead in [SluVág], and recently for a set consisting of deterministic bottom-up tree transformation classes and deterministic top-down tree transformation classes with and without regular look-ahead in [GyeVág].

In this chapter we slightly modify the general method and apply it to the set $M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}$. We chose this particular M because we want to examine how the new class $sl\text{-}DT$ behaves when composing it with known deterministic top-down tree transformation classes.

We note that the results appearing in this chapter were published in the paper [DánFül2].

4.1 The problem and the outline of the solution

In this section we specify the problem that will be solved in the rest of the chapter. Moreover, we present the outline of the solution.

In Section 2.1 we already stated several inclusions and decomposition equations concerning *sl-DT*. Such an inclusion and decomposition are $sl\text{-}DT \subset l\text{-}DT$ and $DT = nd\text{-}HOM \circ sl\text{-}DT$, respectively.

On the other hand, several other inclusions and equations concerning tree transformation classes are known from the literature. For example we recall the inclusion $DT \subset DT^2$ from [Rou] and $HOM \circ l\text{-}DT = DT$ form [Eng3].

We observe that we can obtain some new inclusions and equations from the above ones, e.g. $nd\text{-}HOM \circ sl\text{-}DT \subset DT^2$ and $nd\text{-}HOM \circ sl\text{-}DT = HOM \circ l\text{-}DT$.

This observation motivates us to determine an abstract method, with which all inclusions and equations are derivable that are valid among tree transformation classes obtained from some base tree transformation classes by composition.

Since we are interested in the compositions of *sl-DT* with the well known subclasses of *DT*, we shall choose *HOM*, *sl-DT*, *l-DT*, *nd-DT*, and *DT* as base classes.

We now describe the problem in a more exact way. Let us fix the set

$$M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}.$$

We generalize the problem of inclusion and equality as follows. Whenever given two tree transformation classes

$$X_1 \circ X_2 \circ \ldots \circ X_m$$

and

$$Y_1 \circ Y_2 \circ \ldots \circ Y_n$$

such that $X_i, Y_j \in M$ for $1 \leq i \leq m$ and $1 \leq j \leq n$, we would like to know whether proper inclusion of some direction, equality or incomparability holds between them. Observe that we can answer the question if we can decide whether the inclusion

$$X_1 \circ X_2 \circ \ldots X_m \subseteq Y_1 \circ Y_2 \circ \ldots Y_n$$

holds or not. Really, if we can decide this inclusion, then we can also decide whether

$$Y_1 \circ Y_2 \circ \ldots Y_n \subseteq X_1 \circ X_2 \circ \ldots X_m.$$

Then, for example,

$$X_1 \circ X_2 \circ \ldots X_m \subset Y_1 \circ Y_2 \circ \ldots Y_n$$

if and only if

$$X_1 \circ X_2 \circ \ldots X_m \subseteq Y_1 \circ Y_2 \circ \ldots Y_n$$

and

$$Y_1 \circ Y_2 \circ \ldots Y_n \not\subseteq X_1 \circ X_2 \circ \ldots X_m.$$

So we conclude that we have solved the problem if we present an algorithm which, for any two tree transformation classes as above, decides whether

$$X_1 \circ X_2 \circ \ldots X_m \subseteq Y_1 \circ Y_2 \circ \ldots Y_n$$

holds or not.

We now describe how this algorithm can be developed. Note that a general method was proposed for developing such an algorithm in [FülVág4]. The method presented here is a slight modification of that one.

Observe that the tree transformation classes of the form $X_1 \circ X_2 \circ \ldots \circ X_m$, where $m \geq 0$ and $X_i \in M$, for $1 \leq i \leq m$, constitute a monoid with the composition operation. The identity element of the monoid is I resulting by the empty composition in case $m = 0$. We denote this monoid by $[M]$. Hence our problem is to find an algorithm that decides the inclusion in $[M]$.

We also consider the free monoid M^* generated by M with the concatenation operation, which is denoted by $\cdot$ in this work. The identity relation over M can uniquely be extended to a homomorphism $|\ | : M^* \to [M]$ (see [BurSan]). Then $|\ |$ has the property that, for every element $X_1 \cdot X_2 \cdot \ldots \cdot X_m$ of M^*,

$$|X_1 \cdot X_2 \cdot \ldots \cdot X_m| = X_1 \circ X_2 \circ \ldots \circ X_m$$

holds. In particular, $|e| = I$. Let us denote the kernel of $|\ |$ by θ. Then, certainly, for any two elements $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ of $[M]$, we have

$$X_1 \circ X_2 \circ \ldots \circ X_m = Y_1 \circ Y_2 \circ \ldots \circ Y_n$$

in $[M]$ if and only if

$$|X_1 \cdot X_2 \cdot \ldots \cdot X_m| = |Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n|,$$

or equivalently

$$X_1 \cdot X_2 \cdot \ldots \cdot X_m \,\theta\, Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n$$

in M^*.

Our algorithm is based on the following two corner stones.

1. First we present a confluent and terminating rewriting system $R \subseteq M^* \times M^*$ such that $\Leftrightarrow_R^* = \theta$.

2. Second we present the inclusion diagram of the set

$$|NF(R)| = \{|u| \mid u \in NF(R)\},$$

where $NF(R)$ denotes the set of normal forms of R.

Then, possessing the inclusion diagram for $|NF(R)|$, we can easily decide for any two normal forms $u, v \in NF(R)$ whether $|u| \subseteq |v|$ holds or not.

The algorithm works as follows. Let us be given two elements

$$X_1 \circ X_2 \circ \ldots \circ X_m$$

and

$$Y_1 \circ Y_2 \circ \ldots \circ Y_n$$

of $[M]$. Take the corresponding elements

$$X_1 \cdot X_2 \cdot \ldots \cdot X_m$$

and

$$Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n$$

of M^* and compute the normal forms $u, v \in NF(R)$ such that

$$X_1 \cdot X_2 \cdot \ldots \cdot X_m \overset{*}{\underset{R}{\Rightarrow}} u$$

and

$$Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n \overset{*}{\underset{R}{\Rightarrow}} v,$$

respectively. Since R is terminating and confluent, u and v exist and unique. Moreover,

$$|X_1 \cdot X_2 \cdot \ldots \cdot X_m| = |u|$$

and

$$|Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n| = |v|$$

hold, because $\Rightarrow_R^* \subseteq \Leftrightarrow_R^* = \theta$ and θ is the kernel of $|\ |$. Then, by the definition of $|\ |$, the inclusion

$$X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n \tag{$*$}$$

holds if and only if

$$|X_1 \cdot X_2 \cdot \ldots \cdot X_m| \subseteq |Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n|.$$

On the other hand, this latter inclusion is equivalent to $|u| \subseteq |v|$. Clearly, we can decide by direct inspection of the inclusion diagram whether $|u| \subseteq |v|$ holds or not. Hence we can also decide whether $(*)$ holds or not.

$$
\begin{array}{rll}
(1) & l\text{-}DT^2 \cdot HOM & \rightarrow \; l\text{-}DT \cdot HOM \\
(2) & HOM \cdot HOM & \rightarrow \; HOM \\
(3) & DT \cdot HOM & \rightarrow \; DT^2 \\
(4) & sl\text{-}DT \cdot l\text{-}DT \cdot HOM & \rightarrow \; l\text{-}DT \cdot HOM \\
(5) & l\text{-}DT^3 & \rightarrow \; l\text{-}DT^2 \\
(6) & l\text{-}DT \cdot sl\text{-}DT & \rightarrow \; l\text{-}DT^2 \\
(7) & l\text{-}DT \cdot DT & \rightarrow \; DT^2 \\
(8) & HOM \cdot l\text{-}DT & \rightarrow \; DT \\
(9) & HOM \cdot sl\text{-}DT & \rightarrow \; DT \\
(10) & HOM \cdot DT & \rightarrow \; DT \\
(11) & DT \cdot l\text{-}DT & \rightarrow \; DT^2 \\
(12) & DT \cdot sl\text{-}DT & \rightarrow \; DT^2 \\
(13) & DT^3 & \rightarrow \; DT^2 \\
(14) & sl\text{-}DT \cdot l\text{-}DT^2 & \rightarrow \; l\text{-}DT^2 \\
(15) & sl\text{-}DT \cdot DT^2 & \rightarrow \; DT^2 \\
(16) & nd\text{-}DT \cdot HOM & \rightarrow \; DT^2 \\
(17) & nd\text{-}DT \cdot sl\text{-}DT & \rightarrow \; DT^2 \\
(18) & nd\text{-}DT \cdot l\text{-}DT & \rightarrow \; DT^2 \\
(19) & nd\text{-}DT \cdot nd\text{-}DT & \rightarrow \; nd\text{-}DT \\
(20) & nd\text{-}DT \cdot DT & \rightarrow \; DT^2 \\
(21) & l\text{-}DT \cdot HOM \cdot nd\text{-}DT & \rightarrow \; l\text{-}DT^2 \cdot nd\text{-}DT \\
(22) & DT \cdot nd\text{-}DT & \rightarrow \; DT
\end{array}
$$

Figure 4.1: Rewriting rules of R

4.2 The decidability of inclusions in the composition monoid

Let us fix $M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}$ in the sequel. Recall that θ is the kernel of the homomorphism $|\ | : M^* \to [M]$, as defined in the previous section.

Let the term rewriting system R over M consist of the 22 rewriting rules enumerated in Figure 4.1.

In this section we present an algorithm, which decides the inclusion in $[M]$ in the way described in the previous section. We start by giving a rewriting system R over M^*. Later on, we will show that R is terminating, confluent and that $\Leftrightarrow_R^* = \theta$.

First we show that, informally speaking, the strings in M^*, which are equivalent with respect to $\Leftrightarrow_R^*$, represent the same tree transformation class.

Lemma 4.2.1 *The inclusion $\Leftrightarrow_R^* \subseteq \theta$ holds.*

Proof. To prove the lemma it is sufficient to show that, for every $u \to v \in R$, we have $|u| = |v|$, or equivalently $u\theta v$. In words we say that the elements of R are valid in $[M]$.

Indeed, if the elements of R are valid in $[M]$, then $\Rightarrow_R \subseteq \theta$, which can be seen as follows. Let $w, z \in M^*$ such that $w \Rightarrow_R z$. Then, by the definition of $\Rightarrow_R$, there are strings x and y in M^* and there is a rule $u \to v \in R$ so that $w = x \cdot u \cdot y$ and $z = x \cdot v \cdot y$. Then we can compute as follows:

$$\begin{array}{rcll} |w| & = & |x \cdot u \cdot y| & \\ & = & |x| \circ |u| \circ |y| & \text{(because } |\ | \text{ is a homomorphism)} \\ & = & |x| \circ |v| \circ |y| & \text{(because } |u| = |v|\text{)} \\ & = & |x \cdot v \cdot y| & \text{(because } |\ | \text{ is a homomorphism)} \\ & = & |z|, & \end{array}$$

proving that $w\theta z$.

Analogously, we can show that $\Rightarrow_R^{-1} \subseteq \theta$. Hence $\Leftrightarrow_R \subseteq \theta$ also holds. Finally, being θ the kernel of a homomorphism, we get $\Leftrightarrow_R^* \subseteq \theta^* = \theta$.

So it is enough to show that all elements of R are valid in $[M]$. As a matter of fact, most of them were already proved in earlier works.

For instance, the validity of (8), i.e. that $HOM \circ l\text{-}DT = DT$ was proved in [Bak3] and [Engl]. A lot of the others also follow implicitly from the results of [Bak3] and [Engl]. However, we refer the reader to [FülVágl], because in that latter paper the proofs are explicit.

The validities of (2), (10), (19), and (22) are immediate consequences of Lemma 3 in [FülVágl]. Moreover, (5) and (13) are proved in Consequence 7, (3), (7), (11), (16), (18), and (20) in Lemma 11 of the same paper.

The equations (1) and (21) can be proved using Table 1 in [FülVág1] as follows. From Table 1, it turns out that $l\text{-}DT^2 \circ HOM = l\text{-}nd\text{-}DT \circ HOM$ and also that $l\text{-}DT \circ HOM = l\text{-}nd\text{-}DT \circ HOM$, hence $l\text{-}DT^2 \circ HOM = l\text{-}DT \circ HOM$. The validity of (21) can be shown similarly.

Note that the rest of the rules, namely (4), (6), (9),(12), (14), (15), and (17), all contain the class $sl\text{-}DT$, hence we should present the proofs of validity for them.

We can prove (4) as follows. Since $I \subseteq sl\text{-}DT$, it holds that $l\text{-}DT \circ HOM \subseteq sl\text{-}DT \circ l\text{-}DT \circ HOM$. As for the conversed inclusion, we have $sl\text{-}DT \circ l\text{-}DT \circ HOM \subseteq l\text{-}DT^2 \circ HOM$, because $sl\text{-}DT \subseteq l\text{-}DT$. On the other hand (1) is valid, hence $l\text{-}DT^2 \circ HOM = l\text{-}DT \circ HOM$, which proves the validity of (4).

For (6), we use $l\text{-}DT^2 = l\text{-}DT \circ l\text{-}HOM$ (see Lemma 11 in [FülVág1]). Then $l\text{-}DT^2 = l\text{-}DT \circ l\text{-}HOM \subseteq l\text{-}DT \circ sl\text{-}DT$, because $l\text{-}HOM = sl\text{-}HOM \subseteq sl\text{-}DT$, by Observation 2.1.1. Finally, $l\text{-}DT \circ sl\text{-}DT \subseteq l\text{-}DT^2$. These altogether prove the validity of (6).

The validity of (9) follows from the equation $DT = nd\text{-}HOM \circ sl\text{-}DT$ (see Theorem 2.1.5), from the inclusion $nd\text{-}HOM \subseteq HOM$, and from the fact that homomorphism dt tree transformations are total (see (1) of Proposition 1.4.1).

Next, (12) can be shown quite similarly to the proof of (6), however here we must use the equation $DT^2 = DT \circ l\text{-}HOM$, which comes again from Lemma 11 of [FülVág1].

The validities of (14) and (15) are obvious, because we saw that the equations (5) and (13) are valid.

Finally, (17) can be proved using $DT^2 = nd\text{-}DT \circ l\text{-}HOM$, which was verified in Lemma 11 of [FülVág1]. □

Next we prove that R is terminating. Recall that a weight reducing string rewriting system is necessarily terminating, see Section 1.2.

Lemma 4.2.2 *The string rewriting system R is terminating.*

Proof. A weight function can easily be defined for R so that R is weight reducing. In fact, let $\rho : M \to \{1, 2, \ldots\}$ be such that

$$\begin{aligned} \rho(HOM) &= 3 \\ \rho(sl\text{-}DT) &= 3 \\ \rho(l\text{-}DT) &= 2 \\ \rho(nd\text{-}DT) &= 2 \\ \rho(DT) &= 1. \end{aligned}$$

It is an easy exercise to check that R is weight reducing. Hence it is terminating as well. □

We now give $NF(R)$, i.e. the set of R-normal forms. This happens in three steps. First we specify a set $NF \subseteq M^*$, second we present the inclusion diagram of the tree transformation classes represented by the elements of NF, and finally we show that $NF(R) = NF$.

As for the first step, let NF be defined in the following way.

$$\begin{aligned} NF = \; & \{l\text{-}DT^2, l\text{-}DT \cdot HOM, l\text{-}DT^2 \cdot nd\text{-}DT, DT^2\} \cup \\ & \{sl\text{-}DT^n \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot HOM \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot l\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot nd\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot l\text{-}DT \circ nd\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot DT \mid n \geq 0\} \end{aligned}$$

Observe that $sl\text{-}DT^0 = e$, hence $e \in NF$ holds.

We now present the inclusion diagram of the set of tree transformation classes, which are represented by the elements of NF.

Lemma 4.2.3 *The diagram depicted in Figure 4.2 is the inclusion diagram of the set* $|NF| = \{|u| \mid u \in NF\}$.

Proof. Actually, the involved diagram is a bit more than the inclusion diagram of $|NF|$, because it also contains the suprema of the six hierarchies appearing in $|NF|$. The reason of the insertion of these suprema is that the diagram becomes more complete and easier to handle in this a way.

The proof of the lemma is rather technical and tedious, hence we separated it into Section 4.3. □

By the diagram in Figure 4.2, we have also that each element of NF represents an unique tree transformation class.

Corollary 4.2.4 *For any* $u, w \in NF$, *it holds that* $|u| = |w|$ *if and only if* $u = w$.

Now we are ready to show that the set NF defined above is exactly the set $NF(R)$ of R-normal forms.

Lemma 4.2.5 $NF(R) = NF$

Proof. It is easy, although tedious to show that $NF \subseteq NF(R)$. If we consider the elements of NF one-by-one, then we can realize that there is no element such that a rule in Figure 4.1 is applicable to it.

The proof of $NF(R) \subseteq NF$ is strongly based on the tables depicted in Figures 4.3 and 4.4, which are organized as follows.

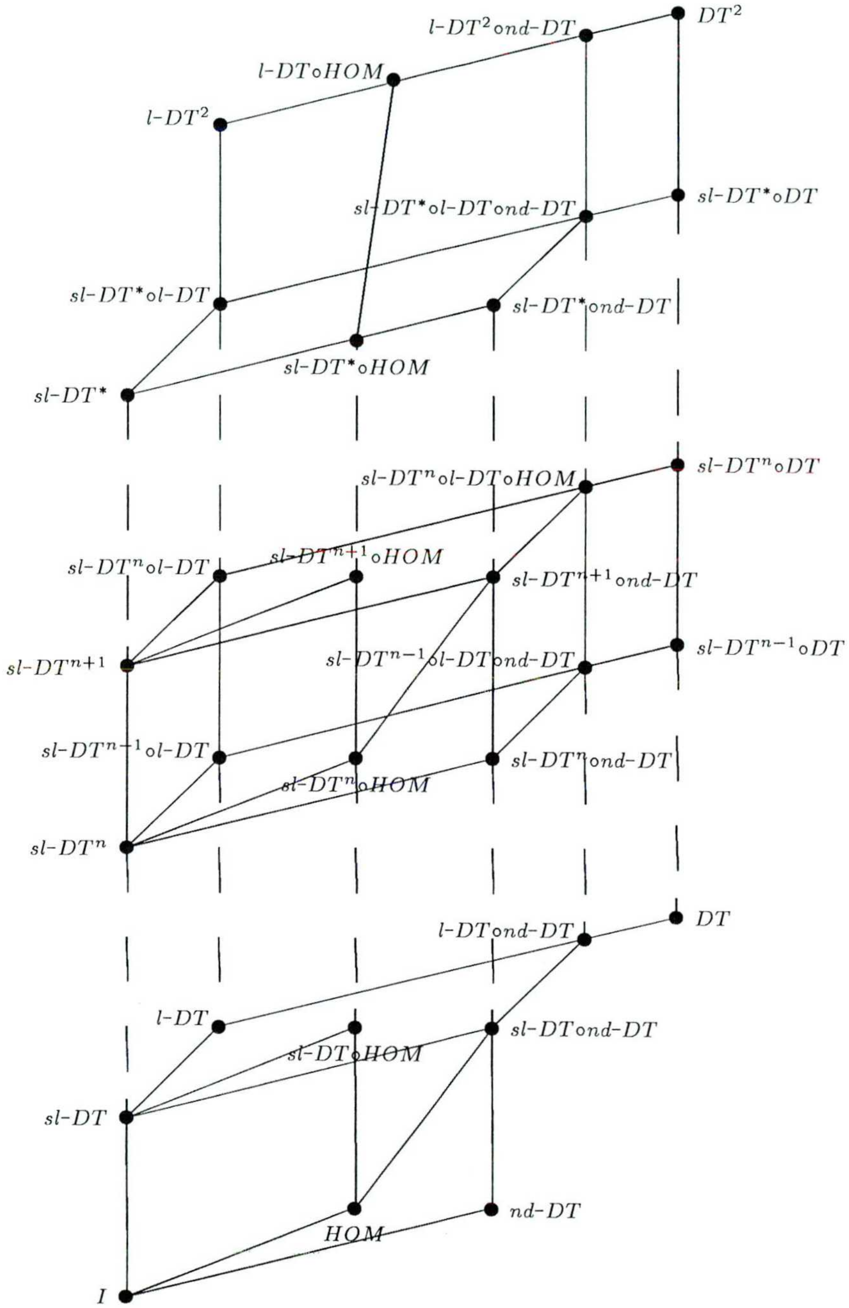

Figure 4.2: The inclusion diagram of normal forms

In fact there is only one table, but it is divided into two parts because of space limitations. Therefore the tables in Figures 4.3 and 4.4 should be considered as one with 10 rows and 5 columns.

The 10 rows are labeled, on the one hand, by the 4 elements of the first union member forming NF and, on the other hand, by the typical elements of the remaining 6 union members constituting NF. Moreover, the columns of the table are labeled by the elements of M.

We now describe how an entry determined by a row labeled by u and a column labeled by C is defined.

If $u \cdot C$ is also an element of NF, then the entry is exactly the word $u \cdot C$ and nothing else.

For example, if $u = sl\text{-}DT^n \cdot HOM$ and $C = nd\text{-}DT$, then the corresponding entry is $sl\text{-}DT^n \cdot HOM \cdot nd\text{-}DT$, because this latter is in NF itself.

However, if $u \cdot C$ is not in NF, then the entry contains also an element v of NF and some number denoting rules in R such that $u \cdot C \Rightarrow_R^* v$ by applying the rules appearing in the entry.

For instance, if $u = sl\text{-}DT^n \cdot l\text{-}DT$ and $C = sl\text{-}DT$, then the entry consists of $l\text{-}DT^2$ and the numbers (6) and (14), because $sl\text{-}DT^n \cdot l\text{-}DT \cdot sl\text{-}DT \Rightarrow_R sl\text{-}DT^n \cdot l\text{-}DT^2$ by equation (6) and $sl\text{-}DT^n \cdot l\text{-}DT^2 \Rightarrow_R^* l\text{-}DT^2$ by applying n times the equation (14).

We now prove that, for every $x \in M^*$, the inclusion $x \in NF(R)$ implies $x \in NF$. The proof is performed by induction on length(x).

Basis. If length$(x) = 0$, then certainly $x = e$. Since $e \in NF$, we have nothing to prove.

Induction step. Now let $x \in M^*$ be such that $length(x) = n + 1$ and suppose that the statement is true for every word in M^* with length at most n. Then $x = y \cdot C$, for some $y \in M^*$ with length n and $C \in M$. Assume that $x \in NF(R)$. Then, certainly, $y \in NF(R)$ (otherwise x could not be in $NF(R)$) and thus, by the induction hypothesis, $y \in NF$, too.

Considering the definition of NF, 10 cases are possible, each of which corresponds to a row in the table, respectively. These are as follows.

Case 1: $y = l\text{-}DT^2$ and $n = 2$. Then we can see from the table that C can only be $nd\text{-}DT$, because in any other cases $x = y \cdot C = l\text{-}DT^2 \cdot C$ can be reduced with some rules of R, hence could not be in $NF(R)$. (For example, in case $C = HOM$, $y \cdot C = l\text{-}DT^2 \cdot HOM$ can be reduced to $l\text{-}DT \cdot HOM$ with rule (1), hence is not irreducible.) However, if $C = nd\text{-}DT$, then $x = y \cdot C = l\text{-}DT^2 \cdot nd\text{-}DT$ is in NF, what we wanted to prove.

Cases 2,3,4: Here $y = l\text{-}DT \cdot HOM$ and $n = 2$, $y = l\text{-}DT^2 \cdot nd\text{-}DT$ and $n = 3$, finally $y = DT^2$ and $n = 2$, respectively. We can see easily from the table that, for every y as above and each $C \in M$, $x = y \cdot C$ cannot be in $NF(R)$, hence we have nothing to prove.

Case 5: $y = sl\text{-}DT^n$. We see from the table that, for every $C \in M$, $x = sl\text{-}DT^n \cdot C$ is in NF.

Since Cases 6–10 can be handled similarly, we left the rest part of the proof for an exercise. □

Now we are ready to show that, roughly speaking, the strings in M^* are equivalent with respect to $\Leftrightarrow_R^*$ if and only if represents the same tree transformation class.

Lemma 4.2.6 *The equality $\theta = \Leftrightarrow_R^*$ holds.*

Proof. Since we have already proved in Lemma 4.2.1 that $\Leftrightarrow_R^* \subseteq \theta$, it is sufficient to show the conversed inclusion $\theta \subseteq \Leftrightarrow_R^*$.

Indeed, let $x, y \in M^*$ be such that $x\theta y$. Since R is terminating (see Lemma 4.2.2) there are words $u, v \in NF(R)$ such that $x \Rightarrow_R^* u$ and $y \Rightarrow_R^* v$.

However, again by Lemma 4.2.1, in this case $x\theta u$ and $y\theta v$ and thus $u\theta v$. On the other hand, by Lemma 4.2.5, u and v are also in NF. Hence, by Corollary 4.2.4, it holds that $u = v$.

We have obtained $x \Rightarrow_R^* u = v \Leftarrow_R^* y$ meaning that $x \Leftrightarrow_R^* y$. This finished the proof of the lemma. □

Finally, we show that any word in M^* can unambiguously be rewritten to an R-normal form.

Lemma 4.2.7 *R is confluent.*

Proof. By Lemma 4.2.2, R is terminating. Thus, by Proposition 1.1.25 in [Jan], it is sufficient to show that every $\Leftrightarrow_R^*$-class contains exactly one R-normal form. (For the proof of this fact, see also [BooOtt].)

Since R is terminating, there is at least one normal form in every $\Leftrightarrow_R^*$-class. Let now $u, v \in NF$ such that $u \Leftrightarrow_R^* v$. Then, by $\Leftrightarrow_R^* = \theta$, we have $u\theta v$. However, by Corollary 4.2.4, this implies also $u = v$. □

Now we are ready to prove the main result of this chapter, i.e. that the inclusion is decidable between any two composition classes consisting of the base tree transformation classes represented by the elements of M.

Theorem 4.2.8 *For any two tree transformation classes $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ in $[M]$, it is decidable whether the inclusion $X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n$ holds.*

Proof. Take the words $x = X_1 \cdot X_2 \cdot \ldots \cdot X_m$ and $y = Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n$. Then, clearly, $|x| = X_1 \circ X_2 \circ \ldots \circ X_m$ and $|y| = Y_1 \circ Y_2 \circ \ldots \circ Y_n$.

	HOM	$sl\text{-}DT$	$l\text{-}DT$
$l\text{-}DT^2$	$l\text{-}DT \cdot HOM$ (1)	DT^2 (6), (5)	DT^2 (5)
$l\text{-}DT \cdot HOM$	$l\text{-}DT \cdot HOM$ (2)	$l\text{-}DT^2$ (9), (7)	$l\text{-}DT^2$ (8), (7)
$l\text{-}DT^2 \cdot nd\text{-}DT$	DT^2 (16), (7), (13)	DT^2 (17), (7), (13)	DT^2 (18), (7), (13)
DT^2	DT^2 (3), (13)	DT^2 (12), (13)	DT^2 (11), (13)
$sl\text{-}DT^n$	$sl\text{-}DT^n \cdot HOM$	$sl\text{-}DT^{n+1}$	$sl\text{-}DT^n \cdot l\text{-}DT$
$sl\text{-}DT^n \cdot HOM$	$sl\text{-}DT^n \cdot HOM$ (2)	$sl\text{-}DT^n \cdot DT$ (9)	$sl\text{-}DT^n \cdot DT$ (8)
$sl\text{-}DT^n \cdot l\text{-}DT$	$l\text{-}DT \cdot HOM$ (4)	$l\text{-}DT^2$ (6), (14)	$l\text{-}DT^2$ (14)
$sl\text{-}DT^n \cdot nd\text{-}DT$	DT^2 (16), (15)	DT^2 (17), (15)	DT^2 (18), (15)
$sl\text{-}DT^n \cdot l\text{-}DT \cdot nd\text{-}DT$	DT^2 (16), (7), (15)	DT^2 (17), (7), (15)	DT^2 (18), (7), (15)
$sl\text{-}DT^n \cdot DT$	DT^2 (3), (15)	DT^2 (12), (15)	DT^2 (11), (15)

Figure 4.3: Table of concatenations with the elements in NF (part 1)

	$nd\text{-}DT$	DT
$l\text{-}DT^2$	$l\text{-}DT^2 \cdot nd\text{-}DT$	DT^2 (7), (13)
$l\text{-}DT \cdot HOM$	$l\text{-}DT^2 \cdot nd\text{-}DT$ (21)	DT^2 (10), (7)
$l\text{-}DT^2 \cdot nd\text{-}DT$	$l\text{-}DT^2 \cdot nd\text{-}DT$ (19)	DT^2 (20), (7), (13)
DT^2	$l\text{-}DT^2 \cdot nd\text{-}DT$ (22)	DT^2 (13)
$sl\text{-}DT^n$	$sl\text{-}DT^n \cdot nd\text{-}DT$	$sl\text{-}DT^n \cdot DT$
$sl\text{-}DT^n \cdot HOM$	$sl\text{-}DT^n \cdot HOM \cdot nd\text{-}DT$	$sl\text{-}DT^n \cdot DT$ (10)
$sl\text{-}DT^n \cdot l\text{-}DT$	$sl\text{-}DT^n \cdot l\text{-}DT \cdot nd\text{-}DT$	DT^2 (7), (15)
$sl\text{-}DT^n \cdot nd\text{-}DT$	$sl\text{-}DT^n \cdot nd\text{-}DT$ (19)	DT^2 (20), (15)
$sl\text{-}DT^n \circ l\text{-}DT \cdot nd\text{-}DT$	$sl\text{-}DT^n \cdot l\text{-}DT \cdot nd\text{-}DT$ (19)	DT^2 (20), (7), (15)
$sl\text{-}DT^n \cdot DT$	$sl\text{-}DT^n \cdot DT$ (22)	DT^2 (15)

Figure 4.4: Table of concatenations with the elements in NF (part 2)

Let u and v be the R-normal forms of x and y, respectively. Since R is terminating and confluent, u and v are unambiguous and can be computed in linear time with respect to length(x) and length(y), just reducing x and y by R, respectively, as long as possible.

Then, by Lemma 4.2.6, also $x\theta u$ and $y\theta v$. Hence $|x| \subseteq |y|$ if and only if $|u| \subseteq |v|$. However, this latter inclusion is obviously decidable by inspecting the inclusion diagram of normal forms depicted in Figure 4.2. □

We finish the section with an example. We would like to know the inclusion relation between the classes

$$sl\text{-}DT^3 \circ HOM \circ l\text{-}DT \circ nd\text{-}DT$$

and

$$sl\text{-}DT^2 \circ HOM \circ sl\text{-}DT \circ HOM \circ nd\text{-}DT.$$

Then we compute as follows:

$$\begin{aligned} sl\text{-}DT^3 \cdot HOM \cdot l\text{-}DT \cdot nd\text{-}DT &\Rightarrow_{R,8} sl\text{-}DT^3 \cdot DT \cdot nd\text{-}DT \\ &\Rightarrow_{R,22} sl\text{-}DT^3 \cdot DT, \end{aligned}$$

where we wrote $\Rightarrow_{R,i}$ to denote that we applied the ith rule in that step of the computation. On the other hand,

$$\begin{aligned} sl\text{-}DT^2 \cdot HOM \cdot sl\text{-}DT \cdot HOM \cdot nd\text{-}DT &\Rightarrow_{R,9} sl\text{-}DT^2 \cdot DT \cdot HOM \cdot nd\text{-}DT \\ &\Rightarrow_{R,3} sl\text{-}DT^2 \cdot DT^2 \cdot nd\text{-}DT \\ &\Rightarrow_{R,22} sl\text{-}DT^2 \cdot DT^2 \\ &\Rightarrow^2_{R,15} DT^2. \end{aligned}$$

Finally, since both $sl\text{-}DT^3 \cdot DT$ and DT^2 are in NF, we can see from the inclusion diagram in Figure 4.2 that $sl\text{-}DT^3 \circ DT \subset DT^2$. Hence the relation $\subset$ holds between the two original classes we started with.

4.3 The inclusion diagram of normal forms

In this section we show that the inclusion diagram of $|NF|$ is exactly the diagram appearing in Figure 4.2.

We recall that $|NF|$ is the set of tree transformation classes represented by NF, which is the set of normal forms in M^* with respect to the string rewriting system R, where M and R are defined at the beginning of Section 4.2.

To present the inclusion diagram we need some technical preparations.

Lemma 4.3.1 *Let $n \geq 0$. Then*

(1) $sl\text{-}DT^n \circ DT \quad = \quad op\text{-}ni\text{-}sl\text{-}DT^n \circ DT$ *and*

(2) $sl\text{-}DT^n \circ nd\text{-}DT \quad = \quad op\text{-}ni\text{-}sl\text{-}DT^n \circ nd\text{-}DT$.

Proof.

(1) Applying n and $n-1$ times (4) and (6) of Corollary 2.1.12, respectively, and finally once (6) of Corollary 1.4.2 to the left-hand side of the equation, we have exactly the right-hand side.

(2) Similarly, we get the right-hand side by applying (4) and (6) of Corollary 2.1.12 and (7) of Corollary 1.4.2 □

We shall also need the following stronger version of (2) of Theorem 3.1.3.

Theorem 4.3.2 $dom(sl\text{-}DT^*) \subset DREC$

Proof. Let $\Sigma = \{\sigma^{(1)}, \#^{(0)}\}$. Define the tree language

$$L = \{\sigma^i(\#) \mid i \geq 0 \text{ is an even integer}\}$$

over Σ. Informally speaking, L is the set of even-length chains over Σ. Note that obviously $L \in DREC$. We prove $L \notin \mathrm{dom}(sl\text{-}DT^*)$.

To see this, suppose the contrary, i.e. that $L \in \mathrm{dom}(sl\text{-}DT^n)$, for some $n \geq 1$. Then there are sl-dt tree transducers $T_1, \ldots, T_n$, such that $L = \mathrm{dom}(\tau_{T_1} \circ \ldots \circ \tau_{T_n})$. We can assume without loss of generality that n is minimal.

Let $T_1 = (Q, \Sigma, \Delta, q, R)$. We investigate the rules in R. Since $\# \in L$, there must be a rule of the form

$$q(\#) \to t_\#$$

in R, for some $t_\# \in T_\Delta$.

The tree $\sigma(\sigma(\#))$ is also in L, hence there must be a (q,σ)-rule in R. It is easy to see that $\mathrm{rhs}(q,\sigma)$ cannot be a ground tree.

On the other hand, T_1 is linear, hence the (q,σ)-rule is of the form $q(\sigma(x_1)) \to t[q'(x_1)]$ in R_1, for some $q' \in Q$ and $t \in \hat{T}_{\Sigma,1}$.

Similarly to the previous argumentation, it is easy to show that there must be a rule of the form $q'(\sigma(x_1)) \to t'[q''(x_1)]$ in R, where $q'' \in Q$ and $t' \in \hat{T}_{\Sigma,1}$.

However, since T_1 is superlinear, this is possible if and only if $q = q' = q''$, meaning that

$$q(\sigma(x_1)) \to t[q(x_1)] \in R.$$

Since $\Sigma = \{\sigma, \#\}$, there cannot be other useful rules in R. We obtained that T_1 is total, which implies $\mathrm{dom}(T_1) = T_\Sigma$. Hence, $n > 1$ must hold.

Consider the above (q,σ)-rule of R. It is easy to see that $t = x_1$ would imply $\tau_{T_1}(\sigma^i(\#)) = t_\#$ for every $i \geq 0$. Hence $t \neq x_1$, meaning that T_1 is nonreducing.

We now have that T_1 is t-nr-sl-dt, that is $L \in \mathrm{dom}(t\text{-}nr\text{-}sl\text{-}DT \circ sl\text{-}DT^{n-1})$, where $n > 1$. Hence, by (3) of Corollary 2.1.12, $L \in \mathrm{dom}(sl\text{-}DT^{n-1})$ holds, which contradicts that n is minimal. □

Moreover, we prove two technical, but very useful lemmas before considering the inclusion diagram of $|NF|$.

We know from Section 3.1 that any sequence of sl-dt tree transducers has "low" transformation power. Roughly speaking, the first lemma shows that the transformation power does not increase significantly even if such a sequence is followed by a dt tree transducer.

Lemma 4.3.3 *Let $L \subset T_\Sigma$ for some ranked alphabet Σ and let $\sigma^{(2)}$, $\#^{(0)}$, and $\$^{(0)}$ be new ranked symbols. Put $\Sigma' = \Sigma \cup \{\sigma^{(2)}, \#^{(0)}, \$^{(0)}\}$ and $\Delta = \{\#^{(0)}, \$^{(0)}\}$. Define the tree transformation $\tau \subseteq T_{\Sigma'} \times T_\Delta$ as*

$$\tau = \{(\sigma(t,s), s) \,|\, t \in L, s \in T_\Delta\}.$$

If $\tau \in sl\text{-}DT^n \circ DT$ for some $n \geq 1$, then $L \in dom(sl\text{-}DT^m)$ holds for some m such that $1 \leq m \leq n$.

Proof. For brevity, we put $K = T_\Delta = \{\#, \$\}$. Observe that $\mathrm{dom}(\tau) = \sigma(L, K)$.

Since $\tau \in sl\text{-}DT^n \circ DT$, the inclusion $\tau \in op\text{-}ni\text{-}sl\text{-}DT^n \circ DT$ holds by (1) of Lemma 4.3.1. That is, op-ni-sl-dt tree transducers $T_1, \ldots, T_n$ and a dt tree transducer T_{n+1} exist such that $\tau = \tau_{T_1} \circ \ldots \circ \tau_{T_n} \circ \tau_{T_{n+1}}$.

Put $T_i = (Q_i, \Delta^{(i-1)}, \Delta^{(i)}, q, R_i)$, where $1 \leq i \leq n+1$. Observe that $\Delta^{(0)} = \Sigma'$ and $\Delta^{(n+1)} = \Delta$. Moreover, we can assume without loss generality that the initial state of all T_i's is q.

Consider the dt tree transducer T_{n+1}. Its output alphabet is Δ, which consists of symbols having rank 0. Therefore, each rule of T_{n+1} either must be a reducing one or it has \$ or # on its right-hand side. That is, T_{n+1} should be an op-ni-l-dt tree transducer. Hence $\tau \in op\text{-}ni\text{-}sl\text{-}DT^n \circ op\text{-}ni\text{-}l\text{-}DT$.

For every $1 \leq i \leq n+1$, we define the type of the sequence $T_1, \ldots, T_i$ by induction on i. This type can be (1), (2) or *undefined.*

(i) The type of T_1 is

- (1) if there is a rule of the form

$$q(\sigma(x_1, x_2)) \to \sigma_1(p_1(x_1), q_1(x_2))$$

in R_1, where $p_1, q_1 \in Q_1$ and $\sigma_1 \in \Delta_2^{(1)}$,

- (2) if there is a rule of the form

$$q(\sigma(x_1, x_2)) \to \sigma_1(q_1(x_2))$$

in R_1, where $q_1 \in Q_1$ and $\sigma_1 \in \Delta_1^{(1)} \cup \{x_1\}$, and

- *undefined* otherwise.

Note that q is the initial state of T_1.

(ii) Let $i \geq 2$. Assume that the type of the sequence $T_1, \dots, T_{i-1}$ has already been defined. The type of $T_1, \dots, T_i$ is

- (1) if the type of $T_1, \dots, T_{i-1}$ is (1) and there is a rule of the form
$$q(\sigma_{i-1}(x_1, x_2)) \to \sigma_i(p_i(x_1), q_i(x_2))$$
in R_i, where $p_i, q_i \in Q_i$ and $\sigma_i \in \Delta_2^{(i)}$,
- (2) if the type of $T_1, \dots, T_{i-1}$ is (1) and there is a rule of the form
$$q(\sigma_{i-1}(x_1, x_2)) \to \sigma_i(q_i(x_2))$$
in R_i, where $q_i \in Q_i$ and $\sigma_i \in \Delta_1^{(i)} \cup \{x_1\}$, and
- <u>*undefined*</u> otherwise.

Note that here q is the initial state of T_i.

We finish the proof as follows. First we make two observations. Observation 1 is on the domains of tree transformations induced by $T_1, \dots, T_i$ sequences defined above. Then, in Observation 2, we characterize the tree transformations induced by sequences of types (1) and (2).

Following this, in Step 1 we show that if there is an integer i with $1 < i \leq n+1$ such that $T_1, \dots, T_i$ is of type (2), then $L = \mathrm{dom}(\tau_{M_1} \circ \dots \circ \tau_{M_{i-1}})$ holds for some sl-dt tree transducers $M_1, \dots, M_{i-1}$.

Finally, in Step 2, we prove that actually there is a sequence $T_1, \dots, T_i$ of type (2), for some $1 < i \leq n+1$.

Observation 1. Consider the tree transducer T_1. Since the root of each input tree in the tree translation τ is σ, there must be a (q, σ)-rule in R_1.

On the other hand, since σ appears only as root in the input trees of τ, we can suppose without loss of generality that this is the only rule containing the state q. (Otherwise, we take a new initial state for T_1.) Then we can also suppose that $\mathrm{dom}(\tau_{T_1}) = \sigma(L_1, K_1)$ holds for some $L_1, K_1 \subseteq T_{\Sigma'}$.

Clearly, $\mathrm{dom}(\tau_{T_1} \circ \dots \circ \tau_{T_{i+1}}) \subseteq \mathrm{dom}(\tau_{T_1} \circ \dots \circ \tau_{T_i})$, for every $1 \leq i \leq n$. Hence we get $\mathrm{dom}(\tau_{T_1} \circ \dots \circ \tau_{T_i}) = \sigma(L_i, K_i)$, for each i such that $1 \leq i \leq n+1$. Moreover, $L_{i+1} \subseteq L_i$, $K_{i+1} \subseteq K_i$, for every $1 \leq i \leq n$. Specially, $L_{n+1} = L$ and $K_{n+1} = K$.

Observation 2. Let $1 \leq i \leq n+1$. If the sequence $T_1, \dots, T_i$ is of type (1), then $\tau_{T_1} \circ \dots \circ \tau_{T_i}$ consists of all pairs of the form $(\sigma(t, s), \sigma_i(t', s'))$, where exist trees $t_0, s_0 \in T_{\Delta^{(0)}}, \dots, t_i, s_i \in T_{\Delta^{(i)}}$ such that

- $t_0 = t$, $s_0 = s$, $t_i = t'$, $s_i = s'$, and,
- for every $1 \leq j \leq i$, $p_j(t_{j-1}) \Rightarrow_{T_j}^* t_j$ and $q_j(s_{j-1}) \Rightarrow_{T_j}^* s_j$

hold. Note that σ_i and the states $p_1, q_1, \ldots, p_i, q_i$ are defined in the definition of the sequence of type (1).

Now suppose that the sequence $T_1, \ldots, T_i$ is of type (2). Then $\tau_{T_1} \circ \ldots \circ \tau_{T_i}$ is the set of all pairs $(\sigma(t,s), \sigma_i(s'))$, where exist trees $t_0, s_0 \in T_{\Delta^{(0)}}, \ldots, t_{i-1}, s_{i-1} \in T_{\Delta^{(i-1)}}, s_i \in T_{\Delta^{(i)}}$ such that

- $t_0 = t$, $s_0 = s$, $s_i = s'$,
- for every $1 \leq j \leq i-1$, $p_j(t_{j-1}) \Rightarrow^*_{T_j} t_j$ and $q_j(s_{j-1}) \Rightarrow^*_{T_j} s_j$, and
- $q_i(s_{i-1}) \Rightarrow^*_{T_i} s_i$

hold.

Step1. By the above observations, it is easy to see that if $T_1, \ldots, T_i$ is of type (2), for some $1 \leq i \leq n+1$, then $\mathrm{dom}(\tau_{T_1} \circ \ldots \circ \tau_{T_i}) = \sigma(L_{i-1}, K_i)$. Moreover, for every j, such that $i \leq j \leq n+1$, $\mathrm{dom}(\tau_{T_1} \circ \ldots \circ \tau_{T_j}) = \sigma(L_{i-1}, K_j)$ should hold.

On the other hand, $\mathrm{dom}(\tau_{T_1} \circ \ldots \circ \tau_{T_{n+1}}) = \mathrm{dom}(\tau) = \sigma(L, K)$, hence we have obtained that $L_{i-1} = L$. This provides that T_1 cannot be of type (2). If it were, then $L = T_{\Sigma'}$ would follow, which is a contradiction.

Now, for each $1 \leq j \leq i-1$, let $M_j = (Q_j, \Delta^{(j-1)}, \Delta^{(j)}, p_j, R_j)$ be constructed from T_j such that we let p_j be the initial state instead of q. (Recall that the sequence $T_1, \ldots, T_{i-1}$ is of type (1), hence there must be a rule of the form $q(\sigma_{j-1}(x_1, x_2)) \rightarrow \sigma_j(p_j(x_1), q_j(x_2))$ in R_j.) By the result of the previous paragraph, in this case $\mathrm{dom}(\tau_{M_1} \circ \ldots \circ \tau_{M_{i-1}}) = L_{i-1} = L$ holds. Note that T_j is a superlinear tree transducer, hence so M_j is.

Step 2. Consider the tree transducer T_1. By the definition of τ, it is obvious that T_1, as a sequence, cannot be of type *undefined*. We have seen that it cannot be of type (2) either, hence it is of type (1).

Let $1 \leq i \leq n$. Suppose that $T_1, \ldots, T_i$ is of type (1). By Observation 2, it is easy to see that the tree transducer T_{i+1} should have a (q, σ_i)-rule. Moreover, by the definition of τ, $\mathrm{rhs}(q, \sigma_i)$ should contain x_2. Otherwise, roughly speaking, T_{i+1} would "loose information" about the second direct subtree of the input tree.

Recall that T_{i+1} is order preserving, nonincreasing and linear, and hence $T_1, \ldots, T_{i+1}$ must be of type (1) or (2).

Finally, we show that the whole sequence $T_1, \ldots, T_{n+1}$ cannot be of type (1). This follows from the fact that the output alphabet Δ^{n+1} of T_{n+1} consists of symbols having rank 0.

Hence there is a sequence $T_1, \ldots, T_i$ of type (2), for some $1 \leq i \leq n+1$. □

Now we apply the previous lemma. To present the inclusion diagram of $|NF|$, we shall need the following results.

Corollary 4.3.4 *Let $n \geq 0$ integer. Then*
(1) $sl\text{-}DT^{n+1} \circ l\text{-}HOM \not\subseteq sl\text{-}DT^n \circ DT$ and
(2) $sl\text{-}DT^{n+2} \not\subseteq sl\text{-}DT^n \circ DT$.

Proof.
(1) Let $L \in \mathrm{dom}(sl\text{-}DT^{n+1}) - \mathrm{dom}(sl\text{-}DT^n)$ (such an L exists by (1) of Theorem 3.1.3). Define τ as in Lemma 4.3.3. It is an easy exercise to show that $\tau \in sl\text{-}DT^{n+1} \circ l\text{-}HOM$. Suppose $\tau \in sl\text{-}DT^n \circ DT$. Then, by Lemma 4.3.3, $L \in \mathrm{dom}(sl\text{-}DT^m)$ holds for some $1 \leq m \leq n$, which is a contradiction. We have that $\tau \notin sl\text{-}DT^n \circ DT$.

(2) Since $l\text{-}HOM \subseteq sl\text{-}DT$ (see Corollary 2.1.2), the statement follows from (1) immediately. □

The next technical lemma shows that an l-dt tree transformation exists, which cannot be induced by a sequence of sl-dt tree transducers followed by an nd-dt tree transducer.

Lemma 4.3.5 *$l\text{-}DT \not\subseteq sl\text{-}DT^* \circ nd\text{-}DT$*

Proof. Let $\Sigma = \{\sigma^{(2)}, \#^{(0)}\}$. Define the l-dt tree transducer

$$T = (\{q_0, q_1, q_2\}, \Sigma, \{\#^{(0)}\}, q_0, R),$$

where R consist of the rules

- $q_0(\sigma(x_1, x_2)) \to q_1(x_1)$,
- $q_1(\sigma(x_1, x_2)) \to q_2(x_1)$,
- $q_2(\sigma(x_1, x_2)) \to q_0(x_2)$, and
- $q_0(\#) \to \#\}$.

Let us investigate the set $\mathrm{dom}(\tau_T)$. (Since the output ranked alphabet of T is $\{\#^{(0)}\}$, one can guess that the proof is actually concerned with domains.)

Define the set $H \subseteq \hat{T}_{\Sigma,1}$ of trees as

$$H = \{\sigma(\sigma(\sigma(t_1, x_1), t_2), t_3) \mid t_1, t_2, t_3 \in T_\Sigma\}.$$

It is easy to check that

$$\mathrm{dom}(\tau_T) = \{h_1[\ldots h_n[\#]\ldots] \mid n \geq 0, h_1, \ldots, h_n \in H\}.$$

Informally speaking, starting from q_0, T steps to the left twice and to the right once on σs, and reaches q_0 again. Moreover, T accepts $\#$ also starting in state q_0. The tree transducer rejects every other tree, which is not in H.

We show that $\tau_T \notin sl\text{-}DT^* \circ nd\text{-}DT$, which implies the lemma immediately. To prove this, suppose the contrary, i.e. that there exist sl-dt tree transducers $T_1, \ldots, T_n$ and an nd-dt tree transducer T_{n+1} such that $\tau_T = \tau_{T_1} \circ \ldots \circ \tau_{T_n} \circ \tau_{T_{n+1}}$. We abbreviate the right-hand side of the previous equation by τ.

Suppose that n is minimal. By (2) of Lemma 4.3.1, it can be assumed that the tree transducers $T_1, \ldots, T_n$ are op-ni-sl-dt.

We observe that T_{n+1} is nondeleting and that, obviously, the tree transformation τ_T cannot be induced without deleting capacity. Hence $n \geq 1$ holds.

Let us assume $T_1 = (Q_1, \Sigma, \Delta, p, R_1)$. (The input alphabet of T_1 can be supposed to be Σ without loss of generality.)

Consider the trees in Figure 4.5. Since $\tau = \tau_T$ is supposed, it is easy to check that the following statement holds.

Statement. $t_1, t_2 \in \mathrm{dom}(\tau)$, $t_3 \notin \mathrm{dom}(\tau)$

We investigate the rules of T_1. Considering Statement and that T_1 is nonincreasing, we have that there must be a rule of the form

$$p(\#) \rightarrow \#_1 \qquad (*)$$

in R_1, where $\#_1 \in \Delta_0$.

By Statement, there must be a (p, σ)-rule in R_1. The tree transducer T_1 is op-ni-sl-dt, hence this rule is of one of the following forms:

(1) $p(\sigma(x_1, x_2)) \rightarrow \sigma_1(p'(x_1))$, where $\sigma_1 \in \Delta_1 \cup \{x_1\}$ and $p' \in Q_1$. In this case, by Statement, it is easy to see that there must be a (p', σ)-rule in R_1. Moreover, $\mathrm{rhs}(p', \sigma)$ must contain x_1, otherwise $t_2 \in \mathrm{dom}(\tau)$ would imply $t_3 \in \mathrm{dom}(\tau)$, which contradicts Statement. By the sl property of T_1, it is possible if and only if $p' = p$. However, in this case $t_2 \in \mathrm{dom}(\tau)$ also implies $t_3 \in \mathrm{dom}(\tau)$. We have that this form is not acceptable for the (p, σ)-rule.

(2) $p(\sigma(x_1, x_2)) \rightarrow \sigma_1(p'(x_2))$, where $\sigma_1 \in \Delta_1 \cup \{x_1\}$ and $p' \in Q_1$. In this case $t_2 \in \mathrm{dom}(\tau)$ implies $t_3 \in \mathrm{dom}(\tau)$, hence this form contradicts Statement as well.

(3) $p(\sigma(x_1, x_2)) \rightarrow \sigma_1(p'(x_1), p''(x_2))$, where $\sigma_1 \in \Delta_2$ and $p', p'' \in Q_1$. We have that this form is the only possible form of (p, σ).

Suppose that $p' \neq p$ in (3). Then, by Statement, there must be a (p', σ)-rule in R_1. By the superlinear property of T_1, $\mathrm{rhs}(p', \sigma)$ must be a ground tree. However, in this case $t_2 \in \mathrm{dom}(\tau)$ implies $t_3 \in \mathrm{dom}(\tau)$, which contradicts Statement. We have obtained $p' = p$.

Now suppose $p' = p$ and $p'' \neq p$. Similarly to the previous observations, one can easily conclude that a (p'', σ)-rule must be in R_1 and $\mathrm{rhs}(p'', \sigma)$ must be a ground tree. But in this case $t_3 \in \mathrm{dom}(\tau)$ follows contradicting Statement.

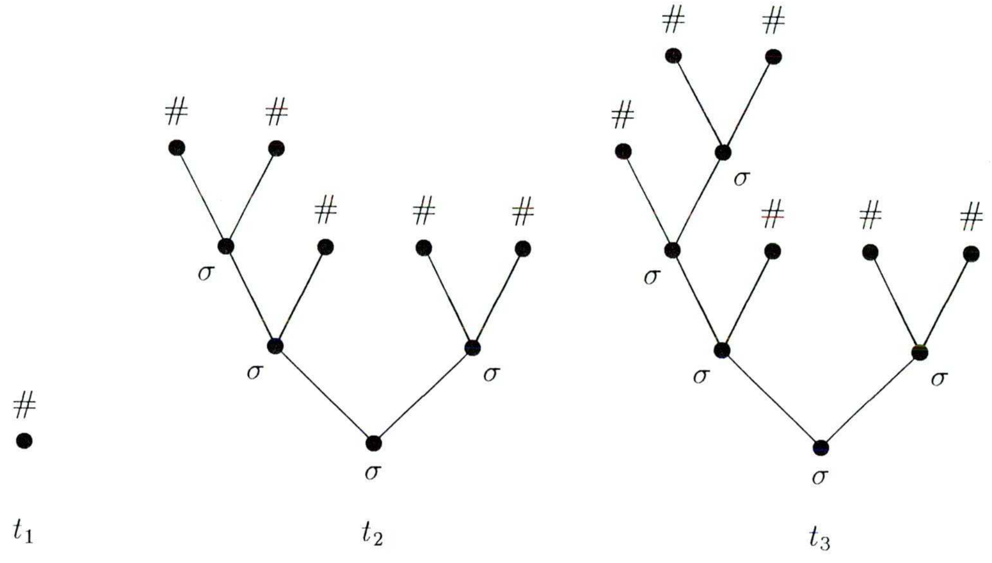

Figure 4.5: Example trees for Lemma 4.3.5

Summarizing up, we have obtained that

$$p(\sigma(x_1, x_2)) \to \sigma_1(p(x_1), p(x_2)) \in R_1. \qquad (**)$$

By the rules $(*)$ and $(**)$, it can be supposed that there are no other rules in R_1. Moreover, $Q_1 = \{p\}$ and $\Delta = \{\sigma_1^{(2)}, \#_1^{(0)}\}$ hold.

We have that T_1 is total and nonreducing, that is t-nr-op-ni-sl-dt tree transducer. Hence, $\tau_T = \tau_{T_1} \circ \ldots \circ \tau_{T_n} \circ \tau_{T_{n+1}}$ implies

$$\tau_T \in t\text{-}nr\text{-}op\text{-}ni\text{-}nd\text{-}sl\text{-}DT \circ op\text{-}ni\text{-}sl\text{-}DT^{n-1} \circ nd\text{-}DT,$$

for some $n \geq 1$.

Assume $n = 1$, then $\tau_T \in t\text{-}op\text{-}ni\text{-}nr\text{-}nd\text{-}sl\text{-}DT \circ nd\text{-}DT = nd\text{-}DT$ holds, which is obviously not true.

Assume $n > 1$, then, by (7) of Corollary 2.1.12, $\tau_T \in op\text{-}ni\text{-}sl\text{-}DT^{n-1} \circ nd\text{-}DT$ follows, which contradicts the minimality of n.

We have that suitable tree transducers $T_1, \ldots, T_{n+1}$ cannot exist. □

Corollary 4.3.6 $l\text{-}DT \circ nd\text{-}DT \not\subseteq sl\text{-}DT^* \circ nd\text{-}DT$

Now we begin to prove Lemma 4.2.3, which states that the diagram depicted in Figure 4.2 is the inclusion diagram of $|NF|$. First we show that all the six hierarchies appearing in $|NF|$ are proper.

Let H be a set of tree transformation classes defined as

$$H = \{I, l\text{-}DT, nd\text{-}DT, HOM, l\text{-}DT \circ nd\text{-}DT, DT\}.$$

Observe that the hierarchies in $|NF|$ are of the form $\{sl\text{-}DT^n \circ X \mid n \geq 0\}$, where $X \in H$. We prove the following.

Lemma 4.3.7 *Let $X \in H$ be arbitrary. Then $\{sl\text{-}DT^n \circ X \mid n \geq 0\}$ is a proper hierarchy.*

Proof. Let $n \geq 0$ and $X \in H$. Recall $sl\text{-}DT^{n+2} \not\subseteq sl\text{-}DT^n \circ DT$ from (2) of Corollary 4.3.4. Since $X \subseteq DT$, we get $sl\text{-}DT^{n+2}X \not\subseteq sl\text{-}DT^n \circ X$.

On the other hand, $sl\text{-}DT^n \circ X \subseteq sl\text{-}DT^{n+2} \circ X$ should be clear. Hence $sl\text{-}DT^n \circ X \subset sl\text{-}DT^{n+2} \circ X$ holds.

Now suppose that $sl\text{-}DT^n \circ X = sl\text{-}DT^{n+1} \circ X$. Then $sl\text{-}DT^{n+1} \circ X = sl\text{-}DT^{n+2} \circ X$ also holds, which implies $sl\text{-}DT^n \circ X = sl\text{-}DT^{n+2} \circ X$. However, this contradicts the result of the previous paragraph.

We have $sl\text{-}DT^n \circ X \subset sl\text{-}DT^{n+1} \circ X$, for every $n \geq 0$ and $X \in H$. □

Let $X \in H$ and consider the classes $sl\text{-}DT^* \circ X = \bigcup_{n \geq 0}(sl\text{-}DT^n \circ X)$, which are the suprema of the corresponding hierarchies. Note that, for every $n \geq 0$, $sl\text{-}DT^n \circ X \subset sl\text{-}DT^* \circ X$ holds by Lemma 4.3.7.

Although the suprema are not elements of $|NF|$, they are very useful to prove certain inclusions in $|NF|$. Moreover, they make the inclusion diagram of $|NF|$ more complete and clear. Therefore, we represented them in the diagram.

In the following lemma we prove the inclusion relations between the suprema of the hierarchies.

Lemma 4.3.8 *The diagram in Figure 4.6 is the inclusion diagram of the set*

$$\{sl\text{-}DT^* \circ X \mid X \in H\},$$

i.e. of the set of suprema of the hierarchies in $|NF|$.

Proof. Observe that all inclusions depicted in Figure 4.6 are obvious, except

$$sl\text{-}DT^* \circ HOM \subseteq sl\text{-}DT^* \circ nd\text{-}DT$$

and

$$sl\text{-}DT^* \circ l\text{-}DT \circ nd\text{-}DT \subseteq sl\text{-}DT^* \circ DT.$$

Hence, to prove the lemma, it is enough to show that the following statements hold:

(1) $sl\text{-}DT^* \circ l\text{-}DT \not\subseteq sl\text{-}DT^* \circ nd\text{-}DT$

(2) $sl\text{-}DT^* \circ HOM \not\subseteq sl\text{-}DT^* \circ l\text{-}DT$

(3) $sl\text{-}DT^* \circ HOM \subset sl\text{-}DT^* \circ nd\text{-}DT$

(4) $sl\text{-}DT^* \circ l\text{-}DT \circ nd\text{-}DT \subset sl\text{-}DT^* \circ DT$

We can prove these statements as follows:

(1) This follows from Lemma 4.3.5 immediately.

(2) Recall $HOM \not\subseteq l\text{-}DT^2$ from Figure 2 of [FülVág3], hence $sl\text{-}DT^* \circ HOM \not\subseteq l\text{-}DT^2$. Since

$$sl\text{-}DT^* \circ l\text{-}DT \subseteq l\text{-}DT^2 \circ l\text{-}DT = l\text{-}DT^2$$

(see (2) of Corollary 2.1.14 and Table 2 of [FülVág1]), the statement holds.

(3) Recall that $HOM = l\text{-}HOM \circ nd\text{-}HOM$ (see (29) in paper [FülVág1]). Since $l\text{-}HOM \subseteq sl\text{-}DT$ holds by Corollary 2.1.2 and $nd\text{-}HOM \subseteq nd\text{-}DT$ is obvious, we have

$$sl\text{-}DT^* \circ HOM \subseteq sl\text{-}DT^* \circ sl\text{-}DT \circ nd\text{-}DT = sl\text{-}DT^* \circ nd\text{-}DT.$$

The hom tree transducers are total, which implies

$$\mathrm{dom}(sl\text{-}DT^* \circ HOM) = \mathrm{dom}(sl\text{-}DT^*) \subset DREC$$

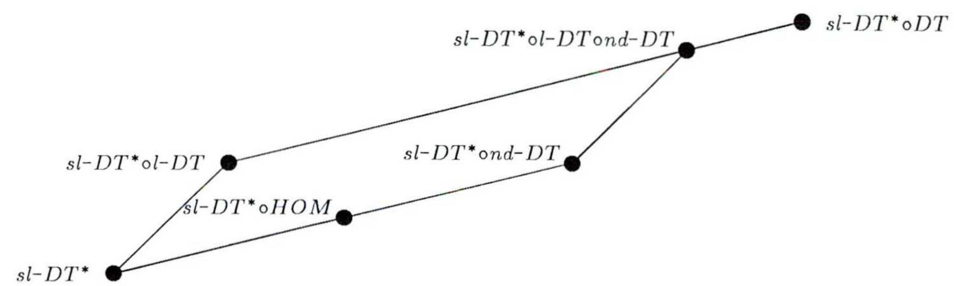

Figure 4.6: The inclusion diagram of suprema

(see Theorem 4.3.2). On the other hand, $\mathrm{dom}(nd\text{-}DT) = DREC$ implies

$$\mathrm{dom}(sl\text{-}DT^* \circ nd\text{-}DT) = DREC,$$

hence the proper inclusion holds.

(4) Since $l\text{-}DT \circ nd\text{-}DT \subseteq DT$ (see Proposition 1.4.1),

$$sl\text{-}DT^* \circ l\text{-}DT \circ nd\text{-}DT \subseteq sl\text{-}DT^* \circ DT$$

holds. Moreover,

$$sl\text{-}DT^* \circ l\text{-}DT \circ nd\text{-}DT \subseteq l\text{-}DT^2 \circ l\text{-}DT \circ nd\text{-}DT \subseteq l\text{-}DT^2 \circ nd\text{-}DT$$

(see (2) of Corollary 2.1.14 and Table 2 of [FülVág1]), and $DT \not\subseteq l\text{-}DT^2 \circ nd\text{-}DT$ (see Figure 2 of [FülVág6]), hence the inclusion is proper.

Observe that the inclusion relation between any two elements depicted in Figure 4.6 can be determined using statements (1)–(4).

For example, we show $sl\text{-}DT^* \subset sl\text{-}DT^* \circ l\text{-}DT$. The inclusions $sl\text{-}DT^* \subseteq sl\text{-}DT^* \circ l\text{-}DT$ and $sl\text{-}DT^* \subseteq sl\text{-}DT^* \circ nd\text{-}DT$ should be obvious. Then, considering (1), we have the desired result immediately. □

Besides the hierarchies, there are the classes $l\text{-}DT^2, l\text{-}DT \circ HOM, l\text{-}DT^2 \circ nd\text{-}DT$ and DT^2 in $|NF|$. In the following lemma we attach them to the inclusion diagram of the suprema of the hierarchies.

Lemma 4.3.9 *The diagram in Figure 4.7 is the inclusion diagram of the set consisting of $l\text{-}DT^2, l\text{-}DT \circ HOM, l\text{-}DT^2 \circ nd\text{-}DT$, DT^2, and the suprema of the six hierarchies.*

Proof. The inclusion relations between the suprema of the hierarchies are clear by Lemma 4.3.8.

In [FülVág6] it has been proved that the four new classes obey the following inclusion relations (see Figure 2 in that paper):

$$l\text{-}DT^2 \subset l\text{-}DT \circ HOM \subset l\text{-}DT^2 \circ nd\text{-}DT \subset DT^2.$$

First we show that none of the suprema (hence none of the elements of the hierarchies) includes any of the new classes. To prove this, it is enough to show that the least new element is not included in the largest supremum, i.e.

$$l\text{-}DT^2 \not\subseteq sl\text{-}DT^* \circ DT \qquad (*)$$

holds.

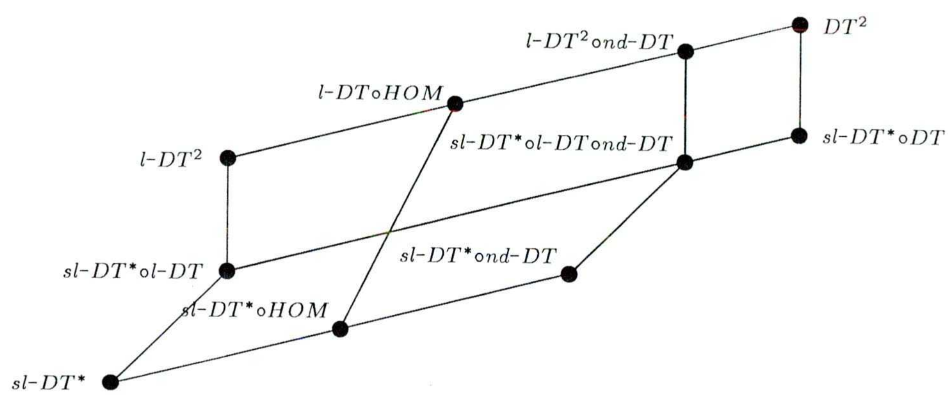

Figure 4.7: The inclusion diagram of suprema and top elements

By Theorem 4.3.2, there exists a tree language L such that $L \in DREC - \mathrm{dom}(sl\text{-}DT^*)$. Construct τ as defined in Lemma 4.3.3. Then it should be clear that $\tau \notin sl\text{-}DT^n \circ DT$ for every $n \geq 0$, hence $\tau \notin sl\text{-}DT^* \circ DT$.

On the other hand, $L \in DREC = \mathrm{dom}(l\text{-}DT)$, therefore $\tau \in l\text{-}DT^2$ should be obvious.

Now we prove that the new elements are the topmost elements in the inclusion diagram of $|NF|$. We define the set G of tree transformation classes as

$$G = \{l\text{-}DT, HOM, l\text{-}DT \circ nd\text{-}DT, DT\}.$$

Note that the set of new four elements of $|NF|$ is exactly $\{l\text{-}DT \circ X \mid X \in G\}$ (for $l\text{-}DT \circ DT = DT^2$ see Table 2 of [FülVág6]).

Let $X \in G$ be arbitrary. Observe that $l\text{-}DT^2 \circ X = l\text{-}DT \circ X$ holds (see Table 2 of [FülVág6]). This implies the inclusion $sl\text{-}DT^* \circ X \subseteq l\text{-}DT \circ X$, because $sl\text{-}DT^* \circ X \subseteq l\text{-}DT^2 \circ X = l\text{-}DT \circ X$ (see (2) of Corollary 2.1.14). Moreover, the inclusion must be proper by $(*)$, that is

$$sl\text{-}DT^* \circ X \subset l\text{-}DT \circ X$$

holds, for each $X \in G$.

Finally, we state that there are no other edges corresponding to the topmost elements in the inclusion diagram of $|NF|$, besides the ones depicted in Figure 4.7. To show this, it is enough to prove the following statements:

(1) $HOM \not\subseteq l\text{-}DT^2$

(2) $nd\text{-}DT \not\subseteq l\text{-}DT \circ HOM$

(3) $DT \not\subseteq l\text{-}DT^2 \circ nd\text{-}DT$

For example, we can show that $sl\text{-}DT^* \circ nd\text{-}DT \not\subseteq l\text{-}DT^2$. For if $sl\text{-}DT^* \circ nd\text{-}DT \subseteq l\text{-}DT^2$ holds, then, by $HOM \subseteq sl\text{-}DT^* \circ HOM$ and $sl\text{-}DT^* \circ HOM \subseteq sl\text{-}DT^* \circ nd\text{-}DT$, we get $HOM \subseteq l\text{-}DT^2$, which contradicts (1).

However, (1), (2), and (3) have already been proved in [FülVág6] (see Figure 2 in that paper). With this, we have proved Lemma 4.3.9. □

We should still prove the inclusions between the elements of the hierarchies. The following corollary shows that, roughly speaking, there can only be edges descending from right to left in the inclusion diagram of $|NF|$.

Corollary 4.3.10 *Denote the bottom elements of the hierarchies as $X_1 = I$, $X_2 = l\text{-}DT$, $X_3 = HOM$, $X_4 = nd\text{-}DT$, $X_5 = l\text{-}DT \circ nd\text{-}DT$ and $X_6 = DT$. Let i, j be arbitrary integers such that $1 \leq i < j \leq 6$. Then $X_j \not\subseteq sl\text{-}DT^* \circ X_i$ holds.*

Proof. By statements (1), (2), and (3) in the proof of Lemma 4.3.9, we have most of these results immediately.

Only the following two cases should be checked:

(1) $l\text{-}DT \not\subseteq sl\text{-}DT^*$ follows from Theorem 2.1.13.

(2) $l\text{-}DT \circ nd\text{-}DT \not\subseteq sl\text{-}DT^* \circ nd\text{-}DT$ holds by Corollary 4.3.6.

Thus we are done. □

Now we can finish the proof of Lemma 4.2.3.

Proof of Lemma 4.2.3. Recall that the inclusion relations between the topmost elements and the suprema of the six hierarchies have already been clarified (see Lemma 4.3.8).

First we prove that the inclusions depicted in Figure 4.2 hold. Let $n \geq 1$. All inclusions should be clear, except the following ones:

(1) $sl\text{-}DT^{n-1} \circ HOM \subseteq sl\text{-}DT^n \circ nd\text{-}DT$

(2) $sl\text{-}DT^{n-1} \circ l\text{-}DT \circ nd\text{-}DT \subseteq sl\text{-}DT^{n-1} \circ DT$

We can prove these statements as follows.

(1) Recall the decomposition equation $HOM = l\text{-}HOM \circ nd\text{-}HOM$ (see (29) in [FülVág1]). Since $l\text{-}HOM \subseteq sl\text{-}DT$ holds by Corollary 2.1.2 and $nd\text{-}HOM \subseteq nd\text{-}DT$ is obvious, we have $sl\text{-}DT^{n-1} \circ HOM \subseteq sl\text{-}DT^{n-1} \circ sl\text{-}DT \circ nd\text{-}DT = sl\text{-}DT^n \circ nd\text{-}DT$.

(2) The inclusion $l\text{-}DT \circ nd\text{-}DT \subseteq DT$ follows from Proposition 1.4.1.

Observe that, by Lemma 4.3.7 and Corollary 4.3.10, the inclusions depicted in Figure 4.2 are necessarily proper.

Finally, we show that there cannot be other inclusions. To prove this, it is enough to consider Corollary 4.3.10 and the following statements:

(3) $l\text{-}DT \not\subseteq sl\text{-}DT^* \circ nd\text{-}DT$, by Lemma 4.3.5.

(4) $sl\text{-}DT^n \circ HOM \not\subseteq sl\text{-}DT^{n-1} \circ DT$, by (1) of Corollary 4.3.4.

(5) $sl\text{-}DT^{n+1} \not\subseteq sl\text{-}DT^{n-1} \circ DT$, by (2) of Corollary 4.3.4.

Now we have obtained that the relations between any two elements depicted in Figure 4.2 can be determined using Corollary 4.3.10 and the statements (1)–(5).

For example, we show

$$sl\text{-}DT^3 \circ HOM \subset sl\text{-}DT^7 \circ nd\text{-}DT.$$

It should be clear that

$$sl\text{-}DT^3 \circ HOM \subseteq sl\text{-}DT^6 \circ HOM.$$

By statement (1), we have

$$sl\text{-}DT^6 \circ HOM \subset sl\text{-}DT^7 \circ nd\text{-}DT.$$

Hence

$$sl\text{-}DT^3 \circ HOM \subset sl\text{-}DT^7 \circ nd\text{-}DT$$

holds.

With this, we finished the proof of the Lemma 4.2.3. □

Conclusions

In this thesis we considered superlinear deterministic top-down tree transducers and the class *sl-DT* of superlinear deterministic top-down tree transformations. Our main results are as follows:

- The classes *sl-DT* and *t-sl-DT* are not closed under the composition.
- $t\text{-}l\text{-}DT - sl\text{-}DT^+ \neq \emptyset$, where $sl\text{-}DT^+$ is the transitive closure of the class *sl-DT* under the composition. Roughly speaking, even the consecutive application of arbitrary many sl-dt tree transducers has no enough transformational power to generate all l-dt tree transformations.
- $DT = nd\text{-}HOM \circ sl\text{-}DT$, that is sl-dt tree transducers have enough computational capacity to generate all dt tree transformations with the help of nondeleting homomorphism tree transducers.
- The class dom(*sl-DT*) is exactly *su-DREC*, i.e. the subclass of *DREC* consisting of those tree languages which are recognized by semi-universal deterministic top-down tree recognizers (su-dttr's).
- For any deterministic recognizable tree language L, it is decidable whether L is in dom(*sl-DT*). Namely, L is in dom(*sl-DT*) if and only if the minimal dttr recognizing L is an su-dttr.
  Thus, being the minimal dttr unique up to the isomorphism, the decision procedure is quite simple. Given a dttr T recognizing L, the dttr T is to be minimalized, then it is decidable by direct inspection whether the resulting minimal dttr is semi-universal.
- The class range(*sl-DT*) is exactly *REC*, that is the class of all recognizable tree languages.
- The hierarchies
  $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 0\}$,
  $\{sl\text{-}DT^n \mid n \geq 0\}$, and
  $\{t\text{-}sl\text{-}DT^n \mid n \geq 0\}$
  are proper.

- We have considered the monoid $[M]$ generated by the tree transformation classes HOM, $sl\text{-}DT$, $l\text{-}DT$, $nd\text{-}DT$, and DT with the composition operation. This is the first work, where $sl\text{-}DT$ is taken as a generator element of a monoid of tree transformation classes.
  Using string rewriting techniques, we have developed an algorithm which, given any two elements $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ of $[M]$, can decide whether the inclusion $X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n$ holds. Of course, in this case it is also decidable whether $\supseteq$, $=$, or incomparability holds. We have represented the elements of $[M]$ by strings, and have presented a terminating and confluent string rewriting system R as well as the inclusion diagram of the normal forms with respect to R.
  The inclusion between two elements of $[M]$ can be decided in the following way. We reduce the strings representing the tree transformation classes $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ to normal forms with respect to R. The string rewriting system R is constructed in such a way that $\subseteq$ (resp. $\supseteq$, $=$, incomparability) holds between the two tree transformation classes if and only if the same relation holds between the tree transformation classes represented by the corresponding normal forms. However, this latter can be read from the inclusion diagram depicted in Figure 4.2.

Finally, we arise two open problems, which may be topic of further research. These are as follows:

Open Problem 1 The superlinear property could easily be defined also for deterministic bottom-up tree transformations, as, e.g., the linearity is defined as well. It would be an interesting task to characterize superlinear deterministic bottom-up tree transducers with respect to similar principles as it done concerning the top-down version in this work.

Open Problem 2 It is known that the equivalence problem of deterministic top-down tree transducers is decidable, see [Ési1]. However, the undecidability of the equivalence problem of nondeterministic top-down tree transducers immediately follows from the fact that the equivalence problem of GSM's is undecidable, see [Gri]. Moreover, this latter result also implies that even the equivalence problem of linear nondeterministic top-down tree transducers is undecidable.
The concept of superlinearity can be generalized to nondeterministic top-down tree transducers in a natural way. It is easy to see that the undecidability of the equivalence problem of superlinear nondeterministic top-down tree transducers does not follow (at least immediately not) from the result of [Gri] on GSM's. Thus decidability questions concerning superlinear nondeterministic top-down tree transducers are worth to be studied. Actually, we guess that the equivalence problem of superlinear nondeterministic top-down transducers is decidable.

Up to now, besides the class of deterministic top-down tree transducers, the equivalence problem has been shown to be decidable only for one subclass of nondeterministic top-down transducers, namely for the class of linear strict letter to-letter top-down tree transducers, see [AndBos]. Note that those tree transducers are the same as our ni-nr-l top-down tree transducers.

Összefoglalás (Summary in Hungarian)

A fatranszfomátorok tulajdonságait a hetvenes évek elejétől kutatják az elméleti számítástudományon belül. Ezek olyan véges eszközök, amelyek rangolt ábécék feletti termeket, más néven fákat képesek feldogozni. Egy fatranszformátor egy fahalmazok feletti bináris relációt, ú.n. fatranszformációt indukál.

A kutatások során számos fatranszformátor tipust vizsgáltak. A top-down fatranszformátor fogalmát Rounds ([Rou]) és Thatcher ([Tha1]) vezette be, a bottom-up változatot pedig Thatcher ([Tha2]). Később a transzformációs kapacitás növelése érdekében bonyolultabb tipusokat is definiáltak, nevezetesen Engelfriet a regulárisan előrenéző top-down fatranszformátort ([Eng2]), Engelfriet és Vogler a makró ([EngVog1]), a high-level ([EngVog2]) és a moduláris ([EngVog3]) fatranszformátort, Fülöp az attribútumos fatranszformátort ([Fül1]), valamint Vogler a high-level moduláris fatranszformátort.

Ebben a dolgozatban csak determinisztikus top-down fatranszformátorokat, ill. ezek által indukált fatranszformációkat vizsgúlunk.

A top-down fatranszformátorok vizsgálatának gyakorlati motivációját az adja, hogy a szintaxis vezérelt fordítóprogramok matematikai modelljeként használhatók (ld. [Eng4]).

A determinisztikus top-down fatranszformátoroknak számos altipusát definiálták. Ebben a disszertációban foglalkozunk többek között lineáris, nemtörlő és homomorfizmus determinisztikus top-down fatranszformátorokkal.

A dolgozatban fatranszformáció osztályon általában meghatározott tipusú fatranszformátorok osztálya által indukált fatranszformációk osztályát értjük. Eszerint tehát definiálhatjuk a determinisztikus top-down fatranszformációk osztályát (DT), valamint annak totális (t-DT), lineáris (l-DT), nemtörlő (nd-DT) és homomorfizmus (HOM) részosztályait. A tipusok szabadon kombinálhatók, így még speciálisabb fatranszformáció osztályok hozhatók létre, például a lineáris nemtörlő fatranszformációk osztálya (l-nd-DT).

Adott tipusú fatranszformátorokat vizsgálva mindig felmerül a kérdés, hogy milyen input fahalmazokat képesek feldolgozni és milyen output fahalmazok jöhetnek létre ezen feldolgozások eredményeként. Egy fatranszformátor lehetséges input, illetve output fahalmazait az általa indukált fatranszfromáció értelmezési tartományának (domain) és értékkészletének (range) hívjuk.

A fahalmazokat fanyelveknek is nevezzük. Hasonlóan a string nyelvekhez, a fanyelvekre is léteznek véges felismerő eszközök (ld. [GécSte4]), amelyek segítségével definiálhatjuk a felismerhető, illetve a determinisztikusan felismerhető fanyelvek osztályait. Kiderült, hogy a determinisztikus top-down fatranszformációk értelmezési tartományai éppen a determinisztikusan felismerhető fanyelvek, továbbá, hogy a lineáris determinisztikus top-down fatranszformációk értékkészleteinek osztálya azonos a felismerhető fanyelvek osztályával.

Mivel a fatranszformációk bináris relációk, a fatranszformációk és fatranszformáció osztályok kompozíciója (jelölése: $\circ$) jól definiált. Ugyanakkor a gyakorlati motiváció szempontjából is igen fontos mind a kompozíció, mind a dekompozíció. A kompozíció tanulmányozásával megtudhatjuk, hogy adott tipusú fatranszformátorokkal végzett többlépcsős fordítás helyettesíthető-e egyetlen fatranszformátor alkalmazásával. A dekompozíció pedig azt mutatja meg, hogy egy adott tipusú fatranszformátor által indukált fordítás elvégezhető-e valamely egyszerűbb tipusok többlépcsős alkalmazásával.

A top-down fatranszformátorokat és fatranszformációkat igen intenzíven tanulmányozták. Az alábbiakban vázlatosan bemutatjuk, hogy milyen irányú kutatások folytak és milyen eredmények születtek ezen a területen.

Úttörő jellegű kutatásaik során Rounds ([Rou]), Thatcher ([Tha1]), Engelfriet ([Eng1], [Eng3]) és Baker ([Bak1], [Bak2],[Bak3]) számos altipust definiáltak (például lineáris, nemtörlő, stb.). Vizsgálták ezek egymáshoz viszonyított transzformációs képességeit, továbbá a kapcsolódó fatranszformáció osztályok néhány zártsági tulajdonságát is megmutatták. Ezen eredményeknek összefoglalása megtalálható [GécSte4]-ben.

A top-down fatranszformációk értékkészleteinek és értelmezési tartományainak felismerhetősége számos kutatót foglalkoztatott, ld. [Rou], [Eng2], [GécSte4], [FülVág1] és [FülVág3].

A top-down fatranszformátorok ekvivalenciájának eldönthetetlensége az általános esetben azonnal adódik Griffiths ([Gri]) eredményéből, mely szerint az általánosított szekvenciális gépek (GSM) ekvivalenciája nem eldönthető. Kiderült azonban, hogy a determinisztikus esetben már eldönthető az ekvivalencia, ld. [Ési1] és [Zac]. Az ekvivalencia eldönthetőségének problémáját néhány egyéb speciális nemdeterminisztikus tipusra is megvizsgálták ([AndBos]). Továbbá, egyéb eldönthetőségi kérdések is tanulmányozásra kerültek (injektivitás, az értékkészlet felismerhetősége, stb.), ld. [Ési1], [Ési2], [Fül4] és [FülGye].

A fatranszformáció osztályok kompozíciós és dekompozíciós tulajdonságainak kutatása igen gyümölcsöző területnek bizonyult. Szinte minden fatranszformátorokkal foglakozó publikáció tartalmaz ilyen eredményt, így kompozíciós és dekompozíciós egyenletek egy igen bő készlete áll rendelkezésre. Ezek kezelhetősége érdekében kívánatossá vált egy egységes megközelítési mód kidolgozása az ilyen jellegű kutatásokra vonatkozóan. Fülöp és Vágvölgyi ([FülVág4], [FülVág6]) javasolt egy általános módszert olyan algoritmus kifejlesztésére, amely fatranszformáció osztályok egy tetszőlegesen rögzített halmaza (bázis halmaz) esetén el tudja dönteni, hogy milyen tartalmazási viszony áll fent a bázis halmaz elemeiből kompozícióval kapott fatranszformáció osztályok között. Az eljárásnak számos alkalmazása ismert, ld. [FülVág4], [FülVág5], [Fül2], [SluVág] és [GyeVág].

Megjegyezzük, hogy magyar nyelven is hozzáférhető néhány fatranszformátorokkal kapcsolatos publikáció, ld. [GécSte1], [GécSte2] és [Fül3].

Jelen dolgozat tárgya a determinisztikus top-down fatranszformátorok egy új tipusának, a szuperlineáris fatranszformátoroknak a vizsgálata. A szuperlineáris determinisztikus top-down fatranszformációk osztályát $sl\text{-}DT$-vel jelöljük, A szuperlineáris determinisztikus top-down fatranszformátorok speciális lineáris determinisztikus top-down fatranszformátorok, továbbá $sl\text{-}DT \subset l\text{-}DT$ is teljesül.

A szuperlineáris determinisztikus top-down fatranszformátorok bevezetését Heiko Vogler javasolta 1992-ben egy Fülöp Zoltánnal folytatott megbeszélés során. A motivációt a jól ismert $DT = nd\text{-}HOM \circ l\text{-}DT$ dekompozíciós egyenlet adta, amely először [Eng1]-ben, illetve [Bak3]-ban jelent meg. Sejtésük szerint a fenti egyenletben $l\text{-}DT$ helyettesíthető a DT osztály egy szűkebb részosztályával is, nevezetesen a szuperlineáris determinisztikus top-down fa-transzformációk osztályával ($sl\text{-}DT$).

A dolgozatban megvizsgáljuk a szuperlineáris determinisztikus top-down fatranszformátorok, ill. a kapcsolódó $sl\text{-}DT$ fatranszformáció osztály tulajdonságait. Bár a kutatás kiindulópontja a $DT = nd\text{-}HOM \circ sl\text{-}DT$ dekompozíciós egyenlet volt, a szuperlineáris determinisztikus top-down fatranszformátorok és fatranszformációk tanulmányozása számos egyéb érdekes eredményt is hozott. A dolgozatban felvetett és megoldott problémák motivációját többnyire a korábbi, fatranszformátorokkal foglalkozó munkák adták, ld. például [Eng1], [Eng3], [Bak3], [FülVág1] és [FülVág2]. A főbb kérdések a következők voltak:

- Milyen tartalmazási relációban áll az $sl\text{-}DT$ osztály a már ismert determinisztikus top-down fatranszformáció osztályokkal, mint például DT-vel, vagy $l\text{-}DT$-vel? Másképpen megfogalmazva, hogyan hasonlítható a szuperlineáris determinisztikus top-down fatranszformátorok fordítási képessége a már ismert tipusokéhoz?

- Zárt-e az $sl\text{-}DT$ osztály a kompozícióra?

- Milyen tipusú fanyelvek lehetnek értelmezési tartományai, illetve értékkészletei szuperlineáris determinisztikus top-down fatranszformációknak?

- Hogyan jellemezhetjük az $sl\text{-}DT$ osztály más ismert fatranszformáció osztályokkal képezett kompozícióit?

A kutatás során olyan eredmények is jelentkeztek, amelyek önmagukban, a szuperlineáris determinisztikus top-down fatranszformátoroktól eltekintve is érdekesek lehetnek. Ilyen például a determinisztikus top-down fafelismerők minimalizálási algoritmusa (ld. az 1.4.4 alfejezetben), a szemiuniverzális determinisztikus top-down fafelismerő definíciója, valamint az a tény, hogy a minimalizálás megőrzi a szemiuniverzális tulajdonságot (ld. a 2.2 fejezetben), továbbá a $DT = op\text{-}ni\text{-}DT \circ nr\text{-}l\text{-}nd\text{-}HOM$ dekompozíciós egyenlet (ld. Lemma 2.1.9).

Megjegyezzük, hogy a dolgozatban leírtak már korábban publikálásra kerül, ld. [DánFül1], [DánFül2] és [Dán].

A dolgozatban található fontosabb eredmények a következők:

- Az $sl\text{-}DT$ és $t\text{-}sl\text{-}DT$ fatranszformáció osztályok nem zártak a kompozícióra.
- $t\text{-}l\text{-}DT - sl\text{-}DT^+ \neq \emptyset$, ahol $sl\text{-}DT^+$ az $sl\text{-}DT$ osztálynak a kompozícióra vonatkozó tranzitív lezártja. Eszerint tehát létezik olyan lineáris determinisztikus fatranszformáció, amelyet nem lehet indukálni szuperlineáris determinisztikus fatranszformátorok semmilyen véges sorozatával.
- $DT = nd\text{-}HOM \circ sl\text{-}DT$, azaz egy tetszőleges determinisztikus top-down fatranszformáció mindig indukálható egy nemtörlő homomorfizmus top-down fatranszformátor és egy szuperlineáris determinisztikus top-down fatranszformátor egymás utáni alkalmazásával.
- A szuperlineáris determinisztikus fatranszformációk értékkészleteinek osztálya ($\mathrm{dom}(sl\text{-}DT)$) éppen a szemiuniverzális determinisztikus top-down fafelismerők (dttr-ek) által felismert fanyelvek osztálya ($su\text{-}DREC$).
- Eldönthető, hogy egy tetszőleges L determinisztikusan felismerhető fanyelv eleme-e a $\mathrm{dom}(sl\text{-}DT)$ fanyelv osztálynak. Nevezetesen, L pontosan akkor eleme a $\mathrm{dom}(sl\text{-}DT)$ osztálynak, ha az L fanyelvet felismerő minimális dttr szemiuniverzális.
  Mivel a minimális dttr az izomorfizmustól erejéig egyértelmű, az eldöntési algoritmus a következő. Meg kell adni egy tetszőleges, L-t felismerő dttr-t, amelyet minimalizálva a kapott minimális dttr-ről a szabályai alapján egyszerű számolással eldönthető, hogy szemiuniverzális-e.
- A $\mathrm{range}(sl\text{-}DT)$ fanyelv osztály azonos a felismerhető fanyelvek osztályával (REC).
- Az alábbi hierarchiák valódiak:
  $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 0\}$,
  $\{sl\text{-}DT^n \mid n \geq 0\}$ és
  $\{t\text{-}sl\text{-}DT^n \mid n \geq 0\}$.
- Megvizsgáltuk a HOM, $sl\text{-}DT$, $l\text{-}DT$, $nd\text{-}DT$ és DT fatranszformáció osztályok, mint generátor elemek által a kompozícióval generált $[M]$ monoidot.

A string átíró rendszerek területéről vett technikák alkalmazásával kifejlesztettünk egy olyan algoritmust, amely tetszőleges $[M]$-beli $X_1 \circ X_2 \circ \ldots \circ X_m$ és $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ elemek esetén eldönti, hogy az $X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n$ tartalmazás teljesül-e. (Ekkor nyilván az is eldönthető, hogy $\supseteq$, $=$, vagy esetleg összehasonlíthatatlanság áll-e fent közöttük.)
Az $[M]$ elemeit stringekkel reprezentáljuk és megadunk egy R konfluens és termináló string átíró rendszert, amelyben az átírási szabályokat kompozíciós és dekompozíciós egyenletekből származtatjuk. Megadjuk továbbá az R-normálformák által reprezentált fatranszformáció osztályok tartalmazási diagramját.
A tartalmazási reláció két fenti alakú $[M]$-beli elem esetén ekkor a következő módon dönthető el. Normálformává redukáljuk R szerint az $X_1 \circ X_2 \circ \ldots \circ X_m$ és az $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ osztályokat reprezentáló stringeket. Az R string átíró rendszer konstrukciójából adódóan a két fatranszformáció osztály között pontosan akkor teljesül $\subseteq$ (illetve $\supseteq$, $=$, vagy összehasonlíthatatlanság), ha ugyanezen reláció áll fent a normálformák által reprezentált osztályok között is. Ez utóbbi azonban könnyen leolvasható a normálformák tartalmazási diagramjáról (ld. Figure 4.2).

Bibliography

[AndBos] Andre, Y. and Bossut, F., Decidability of equivalence for linear letter-to-letter top-down tree transducers, Proceedings of FCT '93 (Z. Ésik ed.), Lecture Notes in Computer Science 710 (1993) 142-151.

[Bak1] Baker, B. S., Tree transducers and families of tree languages, 5th Ann. ACM STC (1973), 200-206.

[Bak2] Baker, B. S., Tree transducers and tree languages, Information and Control 37 (1978), 241-266.

[Bak3] Baker, B. S., Composition of top-down and bottom-up tree transductions, Information and Control 41 (1979) 186-213.

[Boo] Book, R. V., Thue systems and the Church-Rosser Property: replacement systems, specification of formal languages and presentation of monoids, Progress in combinatorics on words (L. Cummings ed.), 1-38, Academic Press, 1983, New York

[BooOtt] Book, R. V. and Otto, F., String-rewriting systems, Springer-Verlag, 1993.

[BurSan] Burris, S. and Sankappanavar, H. P., A course in universal algebra, Springer-Verlag, 1981.

[Dán] Dányi, G., On domain and range tree languages of superlinear deterministic top-down tree transformations, Acta Cybernetica, 12 (1996) 216-277.

[DánFül1] Dányi, G. and Fülöp, Z., Superlinear deterministic top-down tree transducers, Mathematical Systems Theory 29 (1996) 507-534.

[DánFül2] Dányi, G. and Fülöp, Z., Compositions with superlinear deterministic top-down tree transducers, Theoretical Computer Science, accepted for publication.

[Engl] Engelfriet, J., Bottom-up and top-down tree transformations - a comparison, Mathematical Systems Theory 9 (1975) 198-231.

[Eng2] Engelfriet, J., Top-down tree transducers with regular look-ahead, Mathematical Systems Theory 10 (1977) 289-303.

[Eng3] Engelfriet, J., Three hierarchies of transducers, Mathematical Systems Theory 15 (1982) 95-125.

[Eng4] Engelfriet, J., Tree transducers and syntax-directed semantics, Proceedings of the 7th CAAP, Lille, 1982, 95-125.

[EngVog1] Engelfriet, J. and Vogler, H., Macro tree transducers, Journal of Computational System Sciences 31 (1985) 71-146.

[EngVog2] Engelfriet, J. and Vogler, H., High level tree transducers and iterated push-down tree transducers, Acta Informatica 26 (1988) 131-192.

[EngVog3] Engelfriet, J. and Vogler, H., Modular tree transducers, Theoretical Computer Science 78 (1991) 267-303.

[Ési1] Ésik, Z., Decidability results concerning tree transducers I., Acta Cybernetica 5 (1980) 1-20.

[Ési2] Ésik, Z., Decidability results concerning tree transducers II., Acta Cybernetica 6 (1983) 303-314.

[Fül1] Fülöp, Z., On attributed tree transducers, Acta Cybernetica 5 (1981) 261-279.

[Fül2] Fülöp, Z., A complete description for a monoid of deterministic bottom-up tree transformation classes, Theoretical Compututer Science 88 (1991) 253-268.

[Fül3] Fülöp Z., Eldönthetőségi kérdések fatranszformáció osztályok által generált monoidokban, Alkalmazott Matematikai Lapok 15 (1990-91), 219-266.

[Fül4] Fülöp, Z., Undecidable properties of deterministic top-down tree transducers, Theoretical Computer Science 134 (1994) 311-328.

[FülGye] Fülöp, Z. and Gyenizse, P., On injectivity of deterministic top-down tree transducers, Information Processing Letters 48 (1993), 183-188.

[FülVág1] Fülöp, Z. and Vágvölgyi, S., Results on compositions of deterministic root-to-frontier tree transformations, Acta Cybernetica 8 (1987) 49-61.

[FülVág2] Fülöp, Z. and Vágvölgyi, S, An infinite hierarchy of tree transformations in the class NDR, Acta Cybernetica 8 (1987) 153-168.

[FülVág3] Fülöp, Z. and Vágvölgyi, S., On domains of tree transducers, Bulletin of the EATCS 34 (1988) 55-61.

[FülVág4] Fülöp, Z. and Vágvölgyi, S., A finite presentation for a monoid of tree transformation classes, Proceedings of 2nd Conference on Automata, Languages, and Programming Systems (F. Gécseg and I. Peák Eds.), Salgótarján, 1988.

[FülVág5] Fülöp, Z. and Vágvölgyi, S., A complete classification of deterministic root-to-frontier tree transformation classes, Theoretical Compututer Science 81 (1991) 1-15.

[FülVág6] Fülöp, Z. and Vágvölgyi, S., Decidability of the inclusion in monoids generated by tree transformation classes, Tree Automata and Languages (M. Nivat and A. Podelski eds.), Elsevier Science Publishers B. V., Amsterdam, 1992, 381-408.

[GécSte1] Gécseg, F. and Steinby, M., A faautomaták algebrai elmélete I., Matematikai Lapok 26 (1978), 169-207.

[GécSte2] Gécseg, F. and Steinby, M., A faautomaták algebrai elmélete II., Matematikai Lapok 27 (1979), 283-366.

[GécSte3] Gécseg, F. and Steinby, M., Minimal ascending tree automata, Acta Cybernetica 4 (1980) 37-44.

[GécSte4] Gécseg, F. and Steinby, M., Tree automata, Akadémiai Kiadó, Budapest, 1984.

[Gri] Griffiths, T. V., The unsolvability of the equivalence problem for Λ-free nondeterministic generalized machines, Journal of the ACM 15 (1968) 409-413.

[GyeVág] Gyenizse, P. and Vágvölgyi, S., Compositions of deterministic bottom-Up, top-down, and regular look-ahead tree transformations, Theoretical Compututer Science 156 (1996) 71-97.

[Huet] Huet, G., Confluent Reductions: Abstract properties and applications to term rewriting systems, Journal of the Association for Computing Machinery 4 (1980) 797-821.

[Jan] Jantzen, M., Confluent string rewriting, Springer-Verlag, 1988.

[Rou] Rounds, W. C., Mappings on grammars and trees, Mathematical Systems Theory 4 (1970) 257-287.

[SluVág] Slutzki, G. and Vágvölgyi, S., A hierarchy of deterministic top-down tree transformations, Proceedings of FCT '93 (Z. Ésik ed.), Lecture Notes in Computer Science 710 (1993) 440-451.

[Tha1] Thatcher, J. W., Generalized sequential machine maps, Journal of Computational System Sciences 4 (1970) 339-367.

[Tha2] Thatcher, J. W., Tree auttomata: an informal survey, Currents in the Theory of Computing (A. V. Aho ed.) Prenttice-Hall, 1973, 143-172.

[Vog] Vogler, H., High level modular tree transducers, 1978-1988, Ten years IIG, Institutes for Information Processing (H. J. Pongratz and W. Schinnerl eds.), Report Nr. 260, Institutes for Information Processing, Graz University of Technology

[Zac] Zachar, Z., The solvability of the equivalence problem for deterministic frontier-to-root tree transducers, Acta Cybernetica 4 (1979), 167-177.